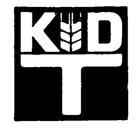



# Einführung in die digitale elektronische Steuerungstechnik

## Lehrbrief 2

KDT-Praktikum



KDT-Praktikum

Einführung in die
digitale elektronische
Steuerungstechnik

# Lehrbrief 2

# Speicherschaltungen und komplexe speicherfreie Standardschaltungen

Autoren: Dr. K. Mauersberger
Dr. D. Fischer

KAMMER DER TECHNIK
PRÄSIDIUM

Fachverband
Elektrotechnik

Sekretariatsbereich
Weiterbildung

Redaktionsschluß: September 1980

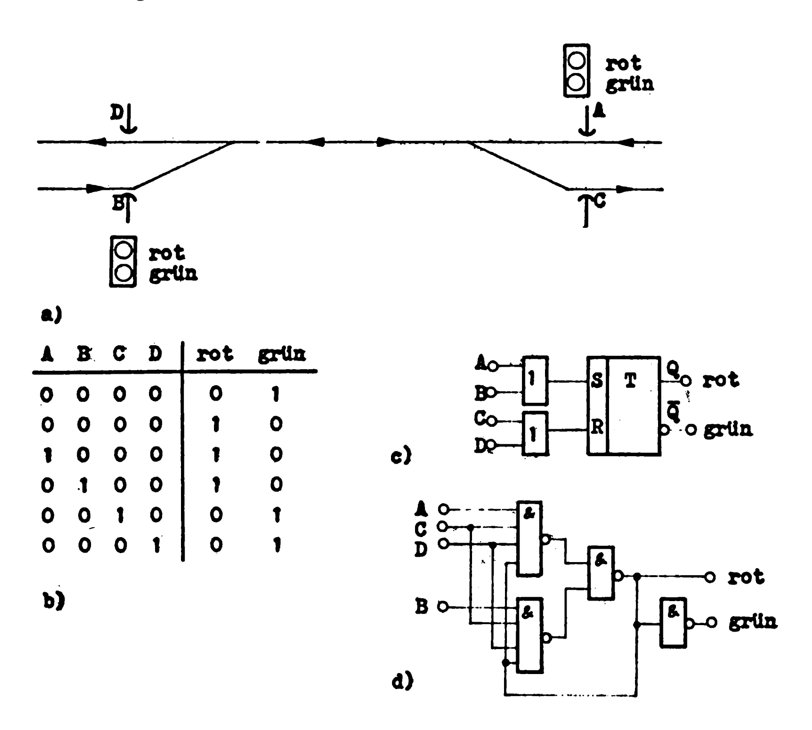
I 12 Ag 238/191/81

Inhaltsverzeichnis Seite

## 4. Schaltungen mit Speicherverhalten

In den vorangehenden Kapiteln haben wir logische Gleichungen und Schaltungen betrachtet, bei denen ein eindeutiger direkter Zusammenhang zwischen den Eingangsvariablen und der Ausgangsgröße bestand. Jeder Kombination der Eingangsvariablen war eindeutig eine Ausgangsgröße (oder mehrere) zugeordnet.

Das folgende Beispiel (Abb. 1) soll zu einer zweiten Gruppe von logischen Funktionen - den Speicherschaltungen oder Sequenzschaltungen - hinführen.




| A | B | C | D | rot | grün |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 0 | 0 | 1 | 0 | 1 |

a) Anlagenskizze
b) Schaltbelegungstabelle
c u.d) Realisierungsbeispiele

Abb. 1 Beispiel einer Ampelsteuerung

Eine längere eingleisige Straßenbahnstrecke ist durch Verkehrsampeln zu regeln. Dabei wollen wir voraussetzen, daß im allgemeinen Rechtsverkehr herrscht und 2 Straßenbahnen nicht genau gleichzeitig in die Strecke einfahren sollen, um die Aufgabe zu vereinfachen. Die Funktionsweise der Anlage kann wie folgt beschrieben werden:

Fährt eine Straßenbahn in die eingleisige Strecke ein, so wird je nach Fahrtrichtung oder Oberleitungskontakt A oder B kurzzeitig betätigt. Damit soll die Ampel auf Rot geschaltet werden, um ein Einfahren einer zweiten Bahn in den Bereich zu verhindern. Wenn die Straßenbahn die eingleisige Strecke wieder verläßt, werden die Oberleitungskontakte C bzw. D kurzzeitig betätigt und dadurch die Ampel wieder auf Grün geschaltet, und eine weitere Bahn kann danach die Stelle passieren.

Beim Entwurf der Steuerung tritt jetzt das Problem auf, daß einmal, wenn kein Oberleitungskontakt betätigt ist und sich keine Bahn in der Engstelle befindet, die Ampel auf Grün steht und zum anderen, wenn ebenfalls kein Oberleitungskontakt betätigt ist und sich eine Bahn in der Engstelle befindet, die Ampel auf Rot steht.

Bei der gleichen Belegung der Eingangsvariablen (Oberleitungskontakte) sind verschiedene Ausgangsfunktionen (Ampelschaltung) zu realisieren.

Diese Aufgabe läßt sich mit den bisherigen Erkenntnissen nicht lösen. Die Schaltung der Ampel ist von vorangegangenen Ereignissen (Kontaktbetätigungen) abhängig, d.h., die Schaltung muß sich "merken", ob vorher die Einfahrt- oder die Ausfahrtkontakte betätigt wurden. Die Elemente, die den vorangegangenen Zustand registrieren und "merken", sind Speicherelemente. In der Relaistechnik wird dieses Verhalten durch die Selbsthalteschaltungen erreicht.

## 4.1. Unterschiede zu kombinatorischen Schaltungen

Die Speicherschaltungen unterscheiden sich von den kombinatorischen Schaltungen im äußeren Verhalten dadurch, daß bei mindestens einer Kombination der Eingangsvariablen verschiedene Ausgangsfunktionen auftreten. In der bekannten Schaltbelegungstabelle führt dies zu mehr als $2^n$ möglichen Kombinationen (n - Anzahl der Eingangsvariablen), um das Gesamtverhalten der Schaltung zu beschreiben. Wenn nicht alle $2^n$ Möglichkeiten genutzt werden, kommt zumindest eine Eingangsbelegung doppelt vor.

Ist die Aufgabe schon als logische Gleichung formuliert, so erkennt man Speicherschaltungen daran, daß die Ausgangsgröße

nicht nur von den Eingangsgrößen, sondern noch von einer inneren Variablen - der Speichergröße Q - abhängt bzw. wenn innere Variable und Ausgangsgröße identisch sind, von sich selbst abhängt.
In Logikplänen sind diese Schaltungen dadurch gekennzeichnet, daß Rückführungen vorhanden sind. Beim Verfolgen der logischen Signale erreicht man mehrmals den gleichen Punkt der Schaltung. Sind Speicherelemente direkt als Symbole im Logikplan vorhanden, wird die Erkennung trivial.
Das Beispiel der Ampelsteuerung zeigt die einzelnen Merkmale deutlich.

## 4.2. Speicherelemente

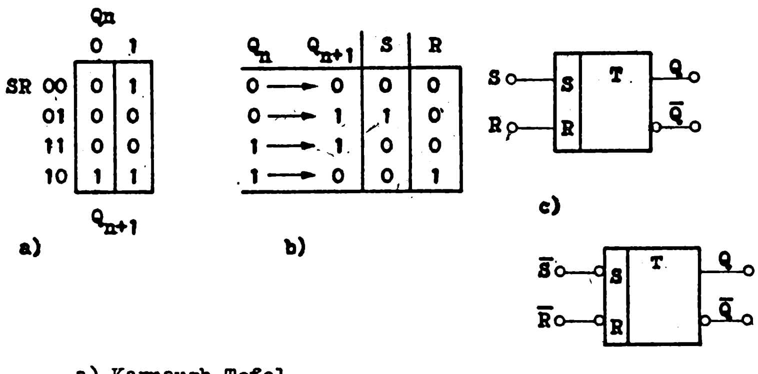
Bei jedem Speicherelement sind drei Betriebsarten interessant:

- Das Setzen des Speichers
  Der Speicher wird durch äußere Signale in den Zustand gebracht, daß das Ausgangssignal gleich Eins wird.
- Das Rücksetzen des Speichers
  Der Speicher wird durch äußere Signale in den Zustand gebracht, daß das Ausgangssignal gleich Null wird.
- Speicherverhalten
  Der Speicher behält den vorher eingeschriebenen Zustand, bis durch ein oben angeführtes Signal eine Änderung vorgenommen wird.

Die Signale Setzen (S) und Rücksetzen (R) werden den Speicherelementen zugeführt und können ihrerseits Ergebnisse logischer Verknüpfungen beliebiger Kompliziertheit sein.

Wir können ein Speicherelement immer in Abhängigkeit von S, R und Q darstellen, wobei S und R von den Eingangsvariablen (und anderen Speichervariablen) abhängen.

Das Verhalten eines Speicherelementes kann in Abhängigkeit von den Setz- und Rücksetzfunktionen in einem Karnaugh-Plan dargestellt werden, wie Abb. 2 zeigt.




a) Karnaugh-Tafel
b) Tabelle der Ansteuerbedingungen
c) Symbol (high-aktiv)
d) Symbol (low-aktiv)

Abb. 2 RS-Flip-Flops

Neu bei dieser Darstellung ist, daß die Speichergröße selbst im Karnaugh-Plan mit erscheint.

Die vierte Zeile kennzeichnet das Setzen des Speichers (nur S = 1). Der Speicher Q wird zwangsweise gesetzt, und deshalb wird in beide Spalten eine Eins eingetragen. War der Speicher vorher schon im Zustand Q = 1, so ändert er sich nicht (2.Spalte). War vorher Q = 0, so wird jetzt Q = 1 erzwungen. Das Rücksetzen wird in der 2. Zeile vorgenommen (nur R = 1). In jedem Falle wird Q = 0 erreicht.

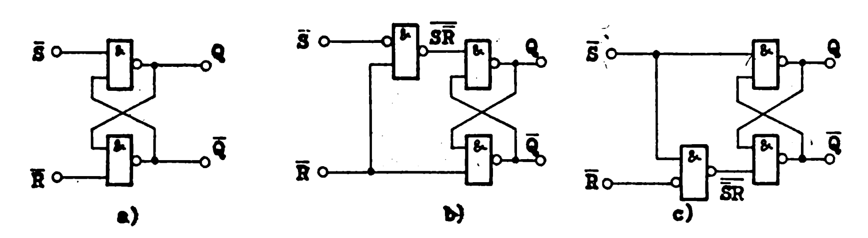
Die erste Zeile schließlich repräsentiert den Zustand Speichern. Es ist weder das Setzsignal noch das Rücksetzsignal gleich Eins, d.h., der Speicher behält seinen vorher eingenommenen Zustand bei. War Q = 0, so bleibt dieser Zustand ebenso wie Q = 1 erhalten. (Die Spalten kennzeichnen ja den vorangegangenen Zustand des Speichers).

In der dritten Zeile des K-Planes sind S = R = 1. Diese Eingangskombination müssen wir zunächst ausschließen, da hier dem Speicherelement widersprüchliche "Anweisungen" erteilt werden.

Die Gleichung für dieses Speicherelement kann in gewohnter Art aus dem Plan herausgelesen werden.

$$Q_{n+1} = S \vee \bar{R}Q_n$$

Das besprochene Speicherelement entspricht einem RS-Flip-Flop (RS - FF) und läßt sich einfach mit NAND-Elementen realisieren. (Abb. 3).



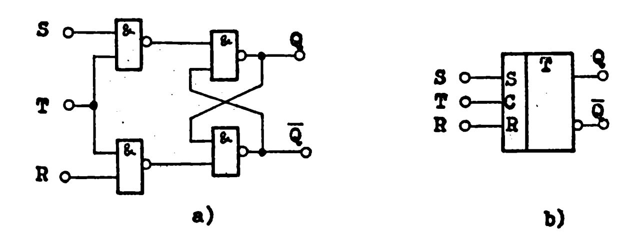

a) RS-FF
b) RS-FF dominierend rückgesetzt
c) RS-FF dominierend gesetzt

Abb. 3 Realisierungsbeispiele für RS-Flip-Flops

Hier wird nochmals der Unterschied zur Relaistechnik deutlich. Bei Relaisschaltungen ist ein gleichzeitiges Auftreten von S und R nicht möglich und demzufolge auch die Möglichkeit, daß Q und $\bar{Q}$ gleichzeitig Eins oder Null werden, nicht gegeben. Bei den Selbsthalteschaltungen dominiert immer S oder R bei gleichzeitiger Signalabgabe. Bei der Realisierung mit NAND-Elementen in der gezeigten Art muß die Dominanz mit zusätzlichen Gattern erzwungen werden.

Nachteilig bei diesen Elementen ist die ungetaktete Arbeitsweise, d.h., jede Signaländerung auf den Leitungen S und R kann den Speicherinhalt (auch ungewollt durch Störungen) verändern.

Diesen Nachteil beseitigt das RST - FF (Abb. 4).




a) Realisierungsbeispiel
b) Symbol

Abb. 4 RST-Flip-Flop

Die Setz- und Rücksetzsignale sind über UND-Gatter (Tore) mit dem Speicherelement verbunden. Die Tore werden in Abhängigkeit von einer weiteren Variablen, dem Taktsignal T, nur geöffnet, wenn das Taktsignal T = 1 ist. Dieser Zustand kann zeitlich sehr kurz sein, so daß die Wahrscheinlichkeit für eine Störung sehr klein wird.

Während des Zustandes T = 0 sind beide Eingänge gesperrt, so daß R = S = 0 gilt und deshalb der gespeicherte Wert erhalten bleibt. Bei einigen Anwendungen besteht die Aufgabe, das auf einer Leitung ankommende Signal (1 Bit Information) getaktet zu speichern. Eine Eins auf dieser Leitung soll also das Setzen und eine Null auf der Leitung soll das Rücksetzen zum Taktzeitpunkt gestatten. Das Rücksetzsignal muß durch Negation erst gewonnen werden. Für diesen Anwendungsfall wurden spezielle Speicher in integrierter Form gefertigt, die als Latch bezeichnet werden. Ihr Vorteil ist u.a. eine niedrige Zahl von Eingängen pro Speicherelement.

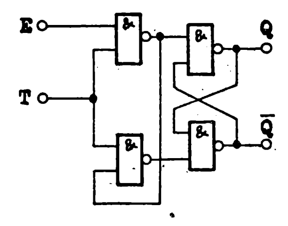



Abb. 5 Realisierungsbeispiel eines Latch

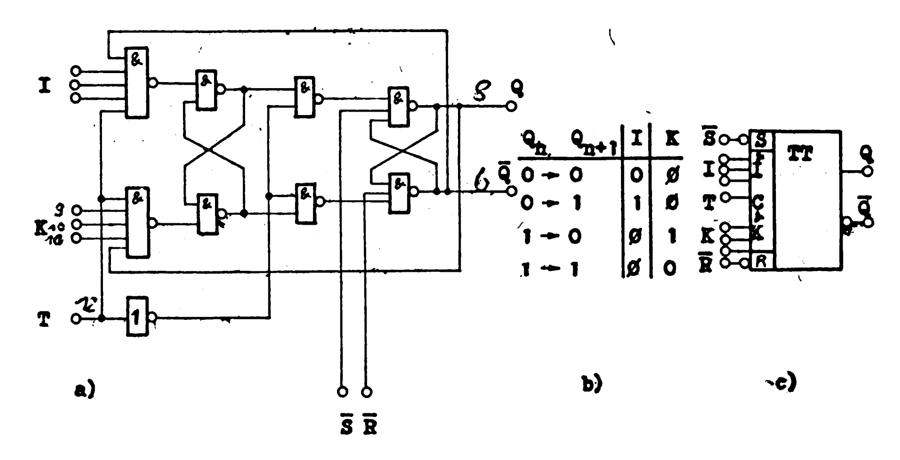
Aus Abb. 5 geht folgende Arbeitsweise hervor. Solange die Tore durch T geöffnet sind, ist Q = E. Wenn das Tor geschlossen wird, bleibt Q auf dem Wert, den E zu diesem Zeitpunkt hatte. Die beschriebenen getakteten Speicher werden für viele Anwendungsfälle eingesetzt. Hauptsächlich finden sie zur Synchronisation mehrerer laufzeitunterscheidlicher Signale Anwendung. Erst nachdem die Daten bzw. Setz- und Rücksetzsignale mit Sicherheit am Speicher anliegen, werden sie durch das Taktsignal in den Speicher übernommen. Ein weiterer wichtiger Anwendungsfall ist der Einsatz zur Entkopplung bei getakteter Arbeitsweise. Durch Zwischenschaltung dieser Speicher kann in einem Takt die logische Verarbeitung der gespeicherten Größe Q erfolgen, und gleichzeitig kann die im nächsten Takt benötigte Information ( z.B. beim Latch) an den Eingang E angelegt werden. Damit kann die durch die Verzögerungszeiten der Bauelemente bedingte Abarbeitungszeit verkürzt werden.

Der Nachteil aller bisher besprochenen Speicherelemente (eine Ausnahme bildet der Sonderfall des Latch) ist, daß S und R nie gleichzeitig Eins werden dürfen. Dieser Nachteil wird durch Rückführungen am Flip-Flop dadurch beseitigt, daß bei gesetztem Speicher über Tore nur noch das Rücksetzsignal wirksam werden kann (bzw. bei rückgesetztem Speicher das Setzsignal).

Die so verknüpften Setz- und Rücksetzsignale werden mit J und K bezeichnet. Ein vielfältig einsetzbares Speicherelement dieses Typs ist das JK-Flip-Flop (D 172). In Abb. 6 ist das logische Verhalten gezeigt. Das Speicherelement besitzt 2 RS-FF, die hintereinander geschaltet sind. Das erste Element wird als Master, der zweite Speicher als Slave bezeichnet.

Man spricht deshalb auch von JK-Master-Slave-FF.

Beide Eingangsgrößen werden für den Master nur mit dem Takt T wirksam, es sind deshalb Setz- bzw. Rücksetzvorbereitungseingänge. Die im Master abgespeicherte Information wird anschließend an den Slave übergeben und dann am Ausgang wirksam.




a) Realisierungsbeispiel eines JK-FF (D 172)
b) Tabelle der Ansteuerungsbedingungungen
c) Symbol

Abb. 6 Realisierungsbeispiel eines JK-FF (D 172)

Die internen Rückführungen von Q und Q̄ bewirken, daß T immer nur an J bzw. K wirksam werden kann, je nachdem ob der Speicher gesetzt war oder nicht. Die einzelnen Phasen während eines Taktsignals zeigt Abb. 7.

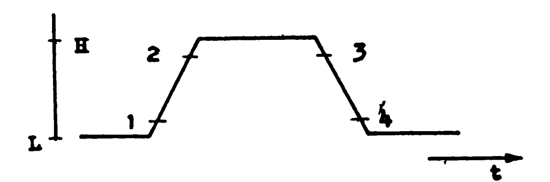



Abb. 7 Takt-Phasenfolge beim JK-FF

Zum Zeitpunkt 1 wird durch Schließen der Tore vor dem Slave die Trennung von Master und Slave erreicht.

Im Zeitraum 2 bis 3 werden die Signale J und K zum Master durchgeschaltet und dieser gesetzt bzw. rückgesetzt. Im Zeitpunkt 3 werden die Tore vor dem Master geschlossen und somit die Trennung der Eingänge vom Master erreicht.

Im Zeitpunkt 4 wird die Information vom Master an den Slave übergeben und damit zu diesem Zeitpunkt am Ausgang wirksam.

Im Baustein D 172 sind jeweils 3 J- und K-Eingänge vorhanden, die intern konjunktiv verknüpft sind. Dadurch wird für viele Anwendungen eine einfache Realisierung der logischen Funktion für J und K ermöglicht. Durch die zusätzlich noch vorhandenen statischen Eingänge S und R, die unabhängig vom Takt wirken, ist dieser Baustein sehr vielseitig einsetzbar.

Eine Beschaltung der J- und K-Eingänge führt zu weiteren typischen Schaltungen.

Bei einer Beschaltung J = K = H arbeitet das JK-FF als Untersetzer. Die an T anliegende Impulsfrequenz wird halbiert. Wie später noch gezeigt wird, lassen sich damit sehr einfach Zähler im Dualkode realisieren.

Wird durch Negation erreicht, daß $K = \overline{J}$ gilt, so arbeitet der Speicher, wenn an J eine Eingangsinformation angelegt wird, als D-FF, wobei die Besonderheit des Master-Slave-Verhaltens berücksichtigt werden muß. Damit können günstig Schieberegister und Verzögerungsglieder aufgebaut werden. Die beschriebenen Speicherelemente können ihre gespeicherte Information nur ändern, wenn der Takt einen bestimmten Zustand (z.B. high) eingenommen hat. Sie werden allgemein als taktzustandsgesteuerte Flip-Flops bezeichnet.

Sie alle besitzen den Nachteil, daß während dieser aktiven Taktphase Änderungen der Eingangsgröße sich auf den Speicherinhalt auswirken.

Dieser Nachteil wird bei den taktflankengesteuerten Flip-Flops beseitigt. Durch Differentiation des Taktimpulses wird die Informationsübernahme praktisch auf die Taktflanke verkürzt. Ein typischer und sehr oft verwendeter Speicher dieser Art ist das D-Flip-Flop.

Dieser Speicher ist ein faktflankengesteuerter Latch. In integrierter Technik ist dieser Speicher als D 174 erhältlich.

Abb. 8 zeigt die logische Schaltung dieses Speichers mit NAND-Elementen.

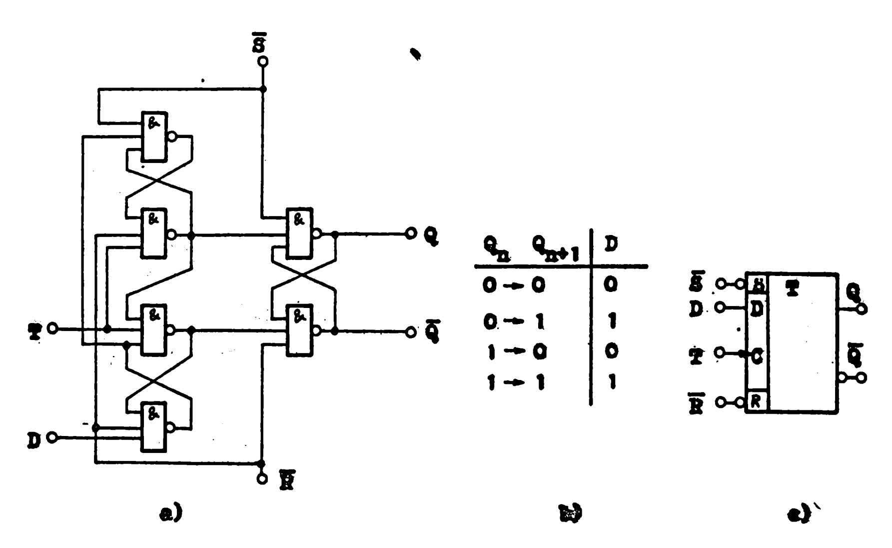



Abb. 8 a) Realisierung eines D-FF (D 174)
b) Tabelle der Ansteuerungsbedingungen
c) Symbol

Anstatt der Differentation des Taktsignals wird bei integrierten Schaltungen der kurze Impuls durch den Laufzeitunterschied eines Gatters (s. auch Lehrbrief 3) gewonnen.

Durch die zusätzlich vorhandenen und ungetaktet wirksamen R- und S-Eingänge ist dieses Bauelement universell einsetzbar.

Für die im Lehrbrief 3 behandelten Parallel- und Schieberegister, Synchronisations- und Verzögerungsschaltungen sind sie das wichtigste Bauelement.

## 4.3. Analyse und Synthese von Speicherschaltungen

Der Entwurf und die Analyse von Speicherschaltungen ist gegenüber den kombinatorischen Schaltungen komplizierter. Neben der Änderung der Eingangsvariablen ist noch der Zustand bzw. die Änderung der Speicherelemente (Zwischenvariablen) zu beachten. Die Zwischenvariablen können eine Instabilität der Gesamtschaltung oder ein "Festlaufen" in einen stationären Zustand bewirken. In beiden Fällen ist die entworfene Schaltung

unbrauchbar.

An Beispielen soll die Analyse und Synthese einfacher Schaltungen demonstriert werden.

### 4.3.1. Analyse von speicherbehafteten Schaltungen

Bei notwendigen Eingriffen in elektronische Baugruppen (Reparaturen, Änderungen der Arbeitsweise...) steht vor dem Wartungspersonal oft die Aufgabe, Teile der gesamten Anlage bezüglich ihrer Arbeitsweise zu analysieren. Dazu steht in der Regel nur der Logikplan zur Verfügung.
Die mit dem Logikplan in Abb. 9 gezeigte Funktion Y soll analysiert werden.

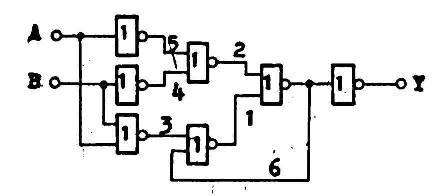



Abb. 9 Logikplan für Analysebeispiel

Zu Beginn muß die Aussage getroffen werden, ob es sich um eine Speicherschaltung handelt.

Beim Aufschreiben der Funktion werden zweckmäßig alle Ausgänge der Gatter gekennzeichnet (hier durch die Ziffern 1 bis 6) und durch Einsetzen die Funktion aufgestellt. Es ergibt sich

$$Y = \overline{\overline{A \vee B} \vee 6} \vee \overline{\overline{A} \vee \overline{B}}$$

Ein weiteres Einsetzen wäre zwecklos, da durch die Rückführung eine Darstellung Y = f (A, B) nicht gelingt. Der Ausgang 6 ist gleichzeitig ein Ausgang des Speichergliedes. Hier wird zweckmäßig die Schaltung aufgeschnitten und symbolisch die Speichergröße $Q_n$ und $Q_n + 1$ angetragen. Damit läßt sich die Speichergleichung aufstellen.

$$Q_{n+1} = \bar{A}\,\bar{B} \vee Q_n (\bar{A} \vee \bar{B}) \qquad Y = \bar{Q}$$

Sie zeigt die für Speicherschaltungen charakteristische Form. Diese Gleichung wird in einen Karnaugh-Plan eingetragen, wobei der Speicherwert Qn als Variable mit angetragen wird und die Felder der Tafel mit den entsprechenden Werten von Qn + 1 belegt werden.

In der Literatur wird diese Tafel oft als Konstituententafel bezeichnet. Ebenso wird die Ausgangsfunktion Y tabuliert (Abb. 10).

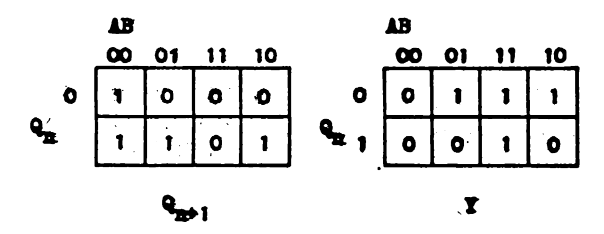



Abb. 10 Konstituententafel zum Analysebeispiel

Dabei werden die Eingangsvariablen vereinbarungsgemäß an den Spalten und die Speichervariablen an den Zeilen angeordnet. Aus der Konstituententafel wird die Wirktafel (Abb. 11) aufgestellt.

| $Q_n$ \ AB | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | 4 | (1) | (2) | (3) |
| 1 | (4) | (5) | 2 | (6) |

Abb. 11 Wirktafel zum Analysebeispiel

In der Wirktafel werden stabile und instabile Zustände unterschieden.

- Bei allen stabilen Zuständen ergibt sich in der Tafel kein Widerspruch zwischen Qn und Qn + 1 (Qn = Qn+1).

Diese stabilen Zustände werden zuerst eingetragen, benannt und mit einem Kreis gekennzeichnet.
Nachdem alle stabilen Zustände eingetragen sind, werden die instabilen Zustände (Qn ≠ Qn + 1) bezeichnet.
Da die Schaltung diese Zustände nur kurzzeitig (nach dem Wechsel der Eingangsvariablen) einnimmt, um dann einen bestimmten stabilen Zustand zu erreichen, wird in diese Felder die Bezeichnung des stabilen Zustandes eingetragen, in den die Schaltung selbständig übergeht.

Aus der Wirktafel kann folgendes erkannt werden:

- Befindet sich in einer Spalte kein stabiler Zustand, so verhält sich die Schaltung bei dieser Eingangsbelegung instabil.
- Befindet sich in keiner Spalte mehr als ein stabiler Zustand, kann die Schaltung ohne Speicherelemente realisiert werden, da jeder Eingangskombination nur eine Ausgangskombination zugeordnet ist.

In einem nächsten Schritt wird die Wirkliste (Abb. 12) aufgestellt. Dabei bleibt die Bezeichnung der Spalten vorteilhaft erhalten. Jeder Zeile wird ein stabiler Zustand zugeordnet. Gleichzeitig werden die Nachbarfelder dieser Zustände durch die Zustände gekennzeichnet, in die die Schaltung übergeht.
Als zusätzliche Spalten werden die Funktionswerte der Ausgangsfunktion für die stabilen Zustände (vergleiche mit K-Plan für Y) aufgenommen. Hiermit kann die Funktionsweise der Schaltung leicht interpretiert werden, bzw. es wird noch ein Impulsdiagramm gezeichnet.

Für unser Beispiel wird deutlich:

Wenn A = B = 1, wird Y = 1.

Dieser Wert bleibt solange erhalten, bis A = B = 0 auftritt. Dann wird Y = 0. Erst wenn A = B = 1 erneut auftritt, wird wieder Y = 1.

AB

| 00 | 01 | 11 | 10 | Y |
|---|---|---|---|---|
| 4 | (1) | 2 |  | 1 |
|  | 1 | (2) | 3 | 1 |
| 4 |  | 2 | (3) | 1 |
| 4 |  | 2 | (6) | 0 |
| 4 | (5) | 2 |  | 0 |
| (4) | 5 |  | 6 | 0 |

Abb. 12 Wirkliste zum Analysebeispiel

Das Beispiel zeigt, daß bei der Analyse allgemein die folgenden Schritte ausgeführt werden sollten:

1. Schritt: Erkennen der Speicherglieder nach Aufstellen der Funktion Qn + 1 = f (A, B, ...Qn)

2. Schritt: Aufstellen der Konstituententafel

3. Schritt: Ableitung der Wirktafel

4. Schritt: Ableitung der Wirkliste

Keiner dieser Schritte enthält eine Änderung der logischen Zusammenhänge. Sie werden nur von verschiedenen Seiten betrachtet. In vielen Fällen kann die Analyse schon nach dem 3. Schritt abgebrochen werden, wenn die Funktionsweise erkannt wurde.

Wenn mehrere Speicherelemente vorhanden sind, entsteht im 1. Schritt ein Funktionenbündel der Form Ql n+1 = f (A, B, ...Q1 n, ...Ql n, ....Qm n)

l = 1 ... m
m = Anzahl der Speicher

Der Übergang vom 2. zum 3. Schritt ist bei Vorhandensein mehrerer Speicherelemente nicht mehr so trivial. Ein stabiler Zustand liegt dann vor, wenn gemäß der genannten Festlegung alle

Speicher einen stabilen Zustand aufweisen.
Im Anhang ist ein Analysebeispiel mit mehreren Speicherelementen aufgezeigt.

### 4.3.2. Synthese von Speicherschaltungen

Bei der Synthese sind prinzipiell die gleichen Schritte wie bei der Analyse auszuführen, allerdings in umgekehrter Reihenfolge. Da für diese Aufgabenstellung viele Schaltungsstrukturen existieren, sind dem Entwerfer bei einigen Schritten Entscheidungen vorbehalten, die zu dieser Vielfalt führen können. Wir wollen wiederum an einem Beispiel die Vorgehensweise erläutern.

Aufgabenstellung:
Ein seriell arbeitendes Addierwerk für Dualzahlen hat als Ausgänge die Summe und den Übertrag. Dieser Übertrag muß in einer Einrichtung gespeichert werden, damit er bei der Verrechnung der nächsthöherwertigen Dualstelle mit berücksichtigt werden kann. Die Operanden seien mit Ak und Bk bezeichnet und der Übertrag mit Yk + 1. Dabei gibt k als Laufindex die Stellenwertigkeit an.

Zur Bildung der Summe wird ein kombinatorisches Netzwerk verwendet, welches hier nicht näher betrachtet werden soll. Für den Übertrag gilt die Tabelle in Abb. 13.

Bei den Kombinationen Ak = 1, Bk = 0 und Ak = 0, Bk = 1 können zwei verschiedene Aussagen zum Übertrag getroffen werden.

War Yk = 0, so wird Yk + 1 = 0. War aber Yk = 1, so bleibt Yk + 1 = 1. Daraus ergeben sich für die gleichen Zustände der Eingangsvariablen verschiedene Zustände der Ausgangsvariablen - also eine Sequenzschaltung.

1. Schritt: Aufstellen der Wirkliste

Die in der Aufgabenstellung enthaltene Abb. 13 zeigt, daß die Schaltung sechs stabile Zustände (entsprechend den Wertekombinationen der Eingangs- und Ausgangsvariablen) besitzt. Diese werden in die Wirkliste eingetragen. Die Spalten entsprechen den Kombinationen der Eingangsvariablen. Jeder Zeile wird ein stabiler Zustand zugeordnet, der wieder wie bei der Analyse gekennzeichnet werden soll.

| $A_K$ | $B_K$ | $Y_{K+1}$ |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 0 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

| $A_K B_K$: 00 | 01 | 11 | 10 | $Y_{K+1}$ |
|---|---|---|---|---|
| (1) | 2 | | 4 | 0 |
| 1 | (2) | 6 | | 0 |
| 1 | (3) | 6 | | 1 |
| 1 | | 6 | (4) | 0 |
| 1 | | 6 | (5) | 1 |
| | 3 | (6) | 5 | 1 |

Abb. 13 Erläuterung zur Syntheseaufgabe und Wirkliste

Nachdem alle stabilen Zustände eingetragen sind, werden diejenigen instabilen Zustände eingetragen, die die Schaltung bei Änderung der Eingangsvariablen durchläuft. Sie werden wieder gleich dem stabilen Zustand benannt, in den die Schaltung übergeht. Nachdem die Wirkliste vollständig ausgefüllt ist, versucht man im zweiten Schritt die Zahl der Zeilen zu reduzieren. Die Zahl der Zeilen entspricht ja wie bei der Analyse der Anzahl der inneren Zustände. Je weniger Zeilen, um so weniger Speicherglieder werden benötigt.

2. Schritt: Reduzierung der Wirkliste

Bei der Zeilenreduzierung versucht man verschiedene Zeilen zu vereinigen. Dies ist möglich, wenn dadurch keine Widersprüche auftreten. Widersprüche in diesem Sinne treten dann auf, wenn in einem Feld der Wirkliste verschiedene Ziffern auftauchen. Die Schaltung müßte dann bei Erreichen dieses Feldes zu verschiedenen stabilen Zuständen übergehen. Dies ist aber bei diesen statischen Schaltungen nicht möglich. Treten keine Widersprüche auf, so können diese Zeilen vereinigt werden. Freie Felder können beliebig belegt werden. Welche Zeilen zweckmäßiger vereinigt werden, ist nicht leicht zu überschauen, wenn mehrere Zustände vorhanden sind.

Eine Entscheidungshilfe bietet das Verträglichkeitsdiagramm (Verschmelzungsdiagramm Abb. 14). Alle stabilen Zustände (Zeilen) werden als Punkt eines Viereckes markiert.

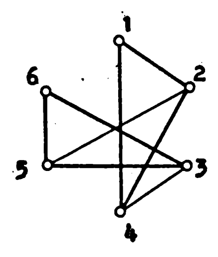



Abb. 14 Verschmelzungsdiagramm

Diejenigen Zeilen, die verschmelzbar sind, werden durch Striche verbunden. Drei Zeilen sind nur verschmelzbar, wenn alle drei widerspruchsfrei sind. Es ergeben sich Dreiecke im Diagramm. Bei vier Zeilen muß für die vollständige Reduzierbarkeit das Viereck mit den Diagonalen entstehen. (Jeder Zustand mit jedem verträglich).

Für das gewählte Beispiel wird Zustand 1, 2, 4 sowie 3, 5, 6 zusammengefaßt. Daraus ergibt sich die reduzierte Wirkliste Abb. 15.

| $A_K B_K$: 00 | 01 | 11 | 10 |
|---|---|---|---|
| (1) | (2) | 6 | (4) |
| 1 | (3) | (6) | (5) |

Abb. 15 Reduzierte Wirkliste zum Synthesebeispiel

Treten stabile und instabile Zustände in einem Feld auf, so wird selbstverständlich der stabile Zustand markiert. Nachdem keine weiteren Möglichkeiten der Reduzierung erkannt werden, ordnet man den Zeilen Zwischenvariablenbelegungen zu. Dabei ist die notwendige Anzahl von Speichergliedern durch die Zeilenzahl festgelegt. Die Anzahl der Speicherglieder ergibt sich aus dem ld der Zeilenzahl. Im Beispiel ist ein Speicher notwendig, da ld 2 = 1.

Werden den Zeilen Speicherzustände zugeordnet, so entsteht aus der Wirkliste die Wirktafel (Abb. 16).

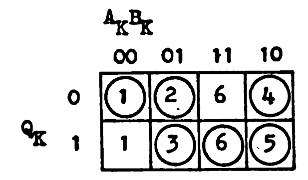



Abb. 16 Wirktafel zum Synthesebeispiel

3. Schritt: Aufstellen der Wirktafel

Die Speichervariable sei Q. Die Zustände werden so gewählt, daß Qk + 1 und Yk + 1 übereinstimmen.
Aus der Wirktafel werden im 4. Schritt die Konstituententafeln der Zwischen- und Ausgangsfunktionen aufgestellt (Abb. 17).

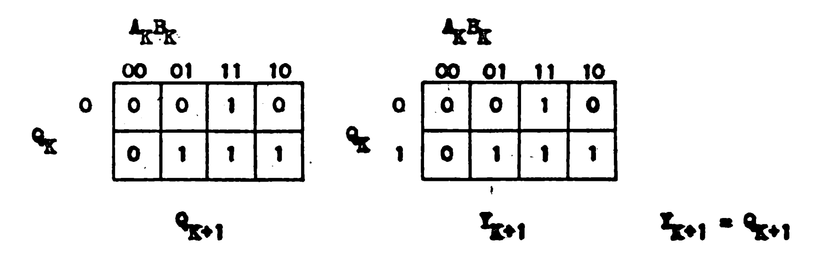



Abb. 17 Konstituententafeln zum Synthesebeispiel

4. Schritt: Aufstellen der Konstituententafeln

Die Eintragung in die einzelnen Felder geschieht wieder so, daß in den stabilen Feldern kein Widerspruch zwischen den Zwischenvariablen und den Zwischenfunktionen besteht.

Die Felder mit instabilen Zuständen werden den gleichnamigen stabilen Zuständen zugeordnet. Gleiche Konstituententafeln werden für Ausgangsfunktionen aufgestellt.

Im 5. Schritt werden die Zwischen- und Ausgangsfunktionen ermittelt. Sie werden aus den Konstituententafeln in gewohnter

Weise herausgelesen.

$$Qk + 1 = Yk + 1 = (Ak \vee Bk)\ Qk \vee AkBk$$

Sollen konzentrierte Speicherelemente verwendet werden, so ist es zweckmäßig, die Setz- und Rücksetzfunktion aus der Zwischenfunktion herauszulesen.
Das Herauslesen der Funktionen S und R geschieht nach folgender Vorschrift:

$$SQ = Qn\ (A,\ B,\ C.....0)$$
$$RQ = \overline{Qn\ (A,\ B,\ C.....1)}$$

D.h., wird in der Speichergleichung Qn = 0 gesetzt, so bleibt nur noch S übrig.
Wird andererseits Qn = 1 gesetzt und die gesamte Funktion negiert, so erhält man R für diesen Speicher.

Für unser Beispiel gilt:

$$S = (\ Ak \vee Bk\ )\ 0 \vee AkBk = AkBk$$
$$R = \overline{(\ Ak \vee Bk\ )\ 1 \vee AkBk}$$
$$R = \overline{Ak \vee Bk} = \overline{Ak}\ \overline{Bk}$$

Daraus kann die in Abb. 18 gezeigte Schaltung entwickelt werden

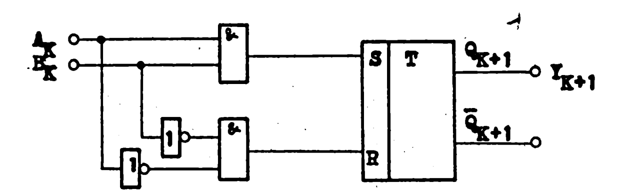



Abb. 18 Schaltung zum Synthesebeispiel

Mit den in den Abbildungen gezeigten Tabellen der Ansteuerbedingungen kann aus der Wirktafel, nachdem ein Speichertyp ausgewählt wurde, gleich der K-Plan für S und R gezeichnet werden. Für ein RS-Flip-Flop ergeben sich die Abb. 19 gezeigten Tafeln. Die Realisierung ist gleich der in Abb. 18 gezeigten Variante.

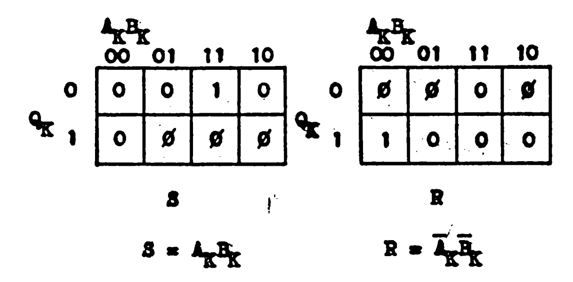



Abb. 19 Tafeln für S und R aus den Ansteuerungsbedingungen für RS-Flip-Flops

Bei Syntheseproblemen, die mit mehr als einem Speicherelement realisiert werden, können sich Probleme ergeben, die in einem weiteren Beispiel betrachtet werden sollen. Als Aufgabenstellung wurde ein gefordertes Impulsdiagramm gewählt, wobei die Änderungen der Eingangsvariablen nicht in konstanten Zeitabschnitten auftreten sollen (t-Achse nicht maßstabsgerecht). Es soll eine Schaltung entworfen werden, die dem in Abb. 20 angegebenen Impulsdiagramm entspricht:

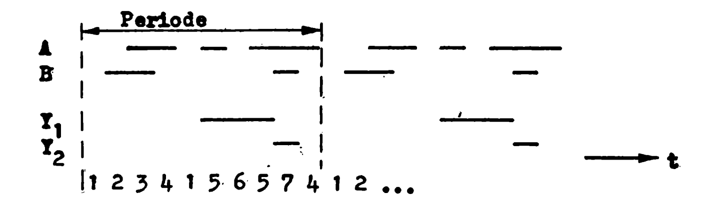



Abb. 20 Impulsdiagramm für 2. Syntheseaufgabe
A, B - Eingangsvariablen
Y1, Y2 - Ausgangsfunktionen

Um aus dieser grafischen Darstellung die Wirkliste aufstellen zu können, werden die verschiedenen Zustände der Schaltung bezeichnet. Dabei werden A, B, Y1 und Y2 betrachtet. Die einzelnen stabilen Zustände sind im Plan gekennzeichnet, wobei gleiche Zustände gleiche Bezeichnungen erhalten. Es ergeben sich 7 Zustände. Diese werden in die Wirkliste eingetragen (Abb. 21).

| AB 00 | 01 | 11 | 10 | $Y_1$ | $Y_2$ |
|---|---|---|---|---|---|
| (1) | 2 | | 5 | 0 | 0 |
| - | (2) | 3 | | 0 | 0 |
| | - | (3) | 4 | 0 | 0 |
| 1 | | - | (4) | 0 | 0 |
| 6 | | 7 | (5) | 1 | 0 |
| (6) | - | | 5 | 1 | 0 |
| | - | (7) | 4 | 0 | 1 |

Abb. 21 Wirkliste zum 2. Synthesebeispiel

Die Auswertung der Aufgabenstellung zeigt, daß nicht alle Nachbarzustände der stabilen Zustände eingenommen werden. So tritt z.B. der Übergang von (2) nach (1) nicht auf. In der Wirkliste sind diese Felder durch Striche gekennzeichnet.

Würde die Wirkliste nicht reduziert werden, so wären zur Realisierung 3 Speicherelemente notwendig, da

$$2 < \mathrm{ld}\ 7 < 3$$

gilt.

Die Reduzierung wird wieder mit dem Verschmelzungsdiagramm vorgenommen (Abb. 22).

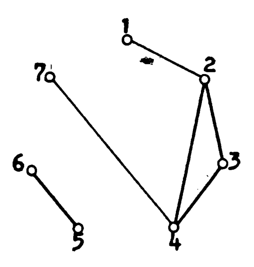



Abb. 22 Verschmelzungsdiagramm für 2. Synthesebeispiel

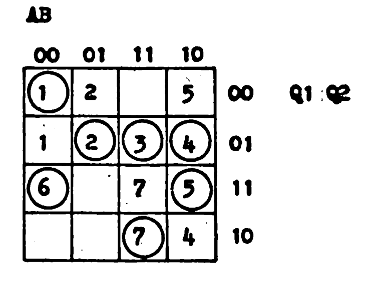
Eine Zusammenfassung der Zeilen 2, 3 und 4 ist offensichtlich. Weiter ist die Verschmelzung von 5 und 6 zu erkennen.
Die Verschmelzung der Zeilen 4 und 7 würde keine Reduzierung der Zeilenzahl ergeben. Dadurch würden aber weniger freie Felder in der reduzierten Wirkliste verbleiben, was beim Aufstellen der Konstituententafel weniger Freizügigkeit bedeutet.

| AB 00 | 01 | 11 | 10 | $Y_1$ | $Y_2$ |
|---|---|---|---|---|---|
| (1) | 2 | | 5 | 0 | 0 |
| 1 | (2) | (3) | (4) | 0 | 0 |
| (6) | | 7 | (5) | 1 | 0 |
| | | (7) | 4 | 0 | 1 |

Abb. 23 Reduzierte Wirkliste zum 2. Synthesebeispiel

Aus der reduzierten Wirkliste (Abb. 23) kann im 3. Schritt die Wirktafel aufgestellt werden (Abb. 24 und Abb. 25).

Der Übergang von der Wirkliste zur Wirktafel ist nicht immer problemlos. Man muß voraussetzen, daß sich immer nur eine Variable ändern kann. Eine gleichzeitige Änderung mehrerer Variablen kann ausgeschlossen werden.



| AB 00 | 01 | 11 | 10 | Q1 Q2 |
|---|---|---|---|---|
| (1) | 2 | | 5 | 00 |
| 1 | (2) | (3) | (4) | 01 |
| (6) | | 7 | (5) | 11 |
| | | (7) | 4 | 10 |

Abb. 24 Wirktafel zum 2. Synthesebeispiel (unbrauchbar)

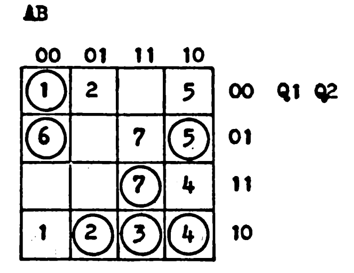
Eine Wirktafel der Form nach Abb. 24 ist unbrauchbar. Die Schaltung kann von (7) bei Änderung der Variablen b von 1 auf 0 nicht sicher den Zustand (4) erreichen, da stabile und instabile Zustände in der Bezeichnung nicht benachbart sind. Es müssen sich also 2 Variable ändern (von 10 auf 01). Da die Speicher nicht exakt gleichzeitig umschalten können, wird der Zwischenzustand 11 oder 00 erreicht.

Der Fall 11 entspricht dem Zustand 5 . Die Schaltung bleibt in diesem Zustand stehen.

Im Falle 00 entspricht dies dem Zustand 5 , was wiederum den Zustand 11 zur Folge haben soll. Als Zwischenwerte treten dabei 01 (ursprüngliches Ziel) oder 10 (Ausgangspunkt der Betrachtung) auf.

Die Schaltung kann zu Schwingungen erregt werden. Eine Beseitigung dieser Unzulänglichkeit kann oft durch Umordnen erreicht werden.

Bei mehr als 4 Zeilen kann dieses Umordnen sehr zeitaufwendig sein. Führt ein Umordnen nicht zum Erfolg, so kann durch Einfügen von Zeilen die Widersprüchlichkeit beseitigt werden. Im Beispiel führt Umordnen zum Ziel. Die neugeordnete Tafel zeigt Abb. 25.



| Q1 Q2 \ AB | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | (1) | 2 | | 5 |
| 01 | (6) | | 7 | (5) |
| 11 | | | (7) | 4 |
| 10 | 1 | (2) | (3) | (4) |

Abb. 25 Wirktafel zum 2. Synthesebeispiel

Jetzt können die Konstituententafeln für Y1, Y2, Q1 und Q2 aufgestellt werden (Abb. 26).
Dabei wird auf die Wirkliste (Abb. 21) zurückgeblickt, und die Tafeln für Y1 und Y2 werden aufgestellt.

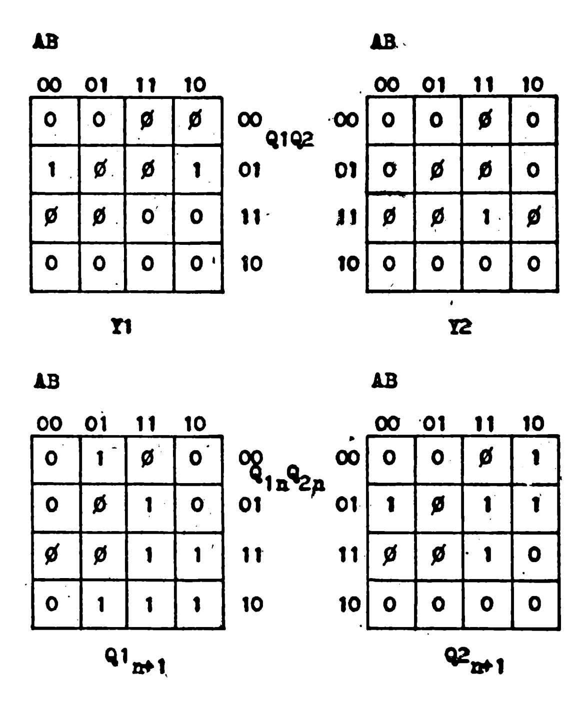



Abb. 26 Konstituententafeln für Y1, Y2, Q1 und Q2

In sehr vielen Fällen ist es für Y1 oder Y2 bedeutungslos, welcher Wert den instabilen Feldern zugeordnet wird, wenn die stabilen Felder den logischen Zustand wechseln. Die flüchtigen Zustände werden in ca. $10^{-8}$s durchlaufen. Dabei ist es für die Ausgangsfunktion oft gleich, ob der Wechsel um diese Zeiten früher oder später vollzogen wird. Es kann dann Ø eingetragen werden.

Freibleibende Felder werden ebenfalls mit Ø gekennzeichnet. Daraus wird abgelesen:

$$Y1 = \overline{Q1}\ Q2 \qquad Y2 = Q1\ Q2$$

$$Q1_{n+1} = B \vee A\, Q1_n \qquad Q2_{n+2} = \overline{Q1}_n\, Q2_n \vee B Q2_n \vee A \overline{Q1}_n$$

Bei einer Realisierung mit konzentrierten Speicherelementen ergeben sich die Funktionen für S und R wie folgt:

$$S_{Q1} = B \qquad S_{Q2} = A\overline{Q1}$$

$$R_{Q1} = \overline{A}\,\overline{B} \qquad R_{Q2} = \overline{B}Q1$$

Damit ist die Synthese beendet und mit den erhaltenen Gleichungen kann die Realisierung der Schaltung vorgenommen werden.

Die Beispiele sollten zeigen, daß auch bei speicherbehafteten Schaltungen systematisch vorgegangen werden kann.

Die hier aufgezeigten Algorithmen werden bei großen Speicherzahlen aufwendig, da durch die Tafeldarstellungen wie beim Karnaugh-Plan bei vielen Variablen die Übersichtlichtkeit verloren geht.

Sie sind deshalb nur für kleine Speicherzahlen vorteilhaft anwendbar. Die Kenntnis dieser Algorithmen erleichtert aber wesentlich das Verständnis komplexer speicherbehafteter Schaltungen, wobei diese oftmals in handhabere Teilschaltungen zerlegbar sind.

## 4.4. Aufgaben und Versuche

1. Stecken Sie die Schaltungen von Abb. 3 auf dem Lehrbaukasten, und machen Sie sich mit der Arbeitsweise vertraut! Überprüfen Sie die Tabelle der Ansteuerbedingungen?
2. Eine dominierend rückgesetzte Selbsthalteschaltung der Relaistechnik soll mit NAND-Elementen realisiert werden. Stellen Sie die logische Gleichung auf, und erprobeh Sie die Schaltung auf dem Lehrbaukasten!
3. In Neubauten wird das Zimmerlicht mittels einfacher Klingeltaster geschaltet. Bei erstmaliger Betätigung des Tasters

wird das Licht ein-, bei nochmaliger Betätigung des gleichen Tasters wird das Licht wieder ausgeschaltet. Entwerfen Sie eine kontaktlose Schaltung dafür, und erproben Sie diese auf dem Lehrbaukasten!

4. Erproben Sie die Schaltung nach Abb. 5 auf dem Lehrbaukasten! Beachten Sie besonders das Verhalten bei Signalwechsel am Eingang E während der aktiven Taktphase!

5. Überprüfen Sie das Verhalten des JK-Flip-Flops (D 172), und vergleichen Sie mit der in Abb. 6 angegebenen Tabelle!

6. Überprüfen Sie das Verhalten eines D-Flip-Flops (D 174), und vergleichen Sie mit der in Abb. 8 angegebenen Tabelle!

7. Welche Änderungen im Verhalten ergeben sich, wenn Sie in Abb. 3a an Stelle der NAND-Elemente NOR-Gatter (bzw. ODER - NICHT) einsetzen?
   Überprüfen Sie Ihre Überlegungen am Lehrbaukasten!

8. Analysieren Sie die in Abb. 27 gezeigte Schaltung!

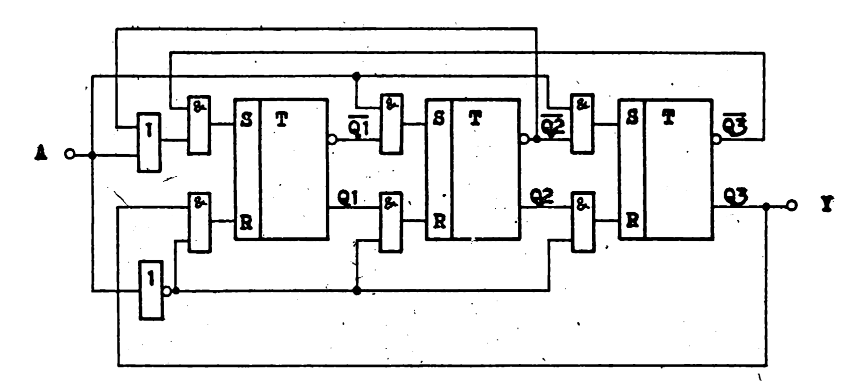



Abb. 27 Logikplan zur Analyseaufgabe

9. Koppeln Sie 2 RST-Flip-Flops, indem Sie Q des ersten Speichers mit S des zweiten Speichers und $\overline{Q}$ mit R verbinden. Untersuchen Sie das Verhalten, wenn Sie

a) die Takteingänge parallel schalten,

b) den Takteingang des zweiten Speichers mit $\overline{T}$ verbinden!

10. Der Füllstand eines Behälters soll zwischen zwei Grenzwerten gehalten werden. Bei Erreichen des unteren Grenzwertes muß die Pumpe eingeschaltet und bei Erreichen des oberen Grenzwertes wieder ausgeschaltet werden. Wenn durch die Flüssigkeitsentnahme der untere Grenzwert wieder erreicht ist, wird erneut zugeschaltet.
Entwerfen Sie die Schaltung, und überprüfen Sie diese am Lehrbaukasten!

11. Ein Stellmotor verstellt über ein Ventil einen für den technologischen Prozeß äußerst wichtigen Stoffstrom.
Für die Betätigung dieses Motors muß das Einverständnis von zwei Überwachungspersonen vorliegen, die räumlich voneinander getrennt sind. Will eine Person den Motor in Betrieb nehmen, so betätigt er eine Taste.
Zunächst soll bei der zweiten Person eine Signallampe aufleuchten. Ist diese mit einer Verstellung einverstanden, quittiert sie dies durch kurzes Drücken auf die Taste.
Daraufhin läuft der Motor so lange, bis die erste Person den Taster nicht mehr betätigt.
Beide Bedienstellen sind gleichberechtigt.
Entwerfen Sie eine Schaltung für diese Aufgabenstellung.
Die Aufgabe kann in folgender Tabelle zusammengefaßt werden:

| A | B | Y1 | Y2 | Y3 | |
|---|---|---|---|---|---|
| A = 0 | B = 0 | Y1 = 0 | Y2 = 0 | Y3 = 0 | |
| A = 1 | B = 0 | Y1 = 0 | Y2 = 1 | Y3 = 0 | vor Quittierung |
| A = 1 | B = 1 | Y1 = 0 | Y2 = 0 | Y3 = 1 | |
| A = 1 | B = 0 | Y1 = 0 | Y2 = 0 | Y3 = 1 | nach Quittierung |
| A = 0 | B = 1 | Y1 = 1 | Y2 = 0 | Y3 = 0 | vor Quittierung |
| A = 1 | B = 1 | Y1 = 0 | Y2 = 0 | Y3 = 1 | |
| A = 0 | B = 1 | Y1 = 0 | Y2 = 0 | Y3 = 1 | nach Quittierung |

A, B - Taster zur Bedienung
Y1 - Lampe bei Bedienstelle A
Y2 - Lampe bei Bedienstelle B
Y3 - Stellmotorensignal

12. Entwerfen Sie einen Dualzähler, der Signalwechsel der Eingangsgröße A zählt.

Als Zählbereich soll durch 2 Speicherelemente der Bereich 0 bis 3 ausreichen. Der 4. Impuls wird wieder mit 0 gezählt.

## 5. Komplexe Standardschaltungen

Nachdem wir in den vorangegangenen Kapiteln - ausgehend von den Bauelementen - die kombinatorischen und sequentiellen Elementarschaltungen behandelt haben, wenden wir uns jetzt umfangreicheren Anordnungen zu. Erst mit diesen Kenntnissen ausgerüstet werden wir in der Lage sein, die in der Praxis auftretenden Aufgabenstellungen zu lösen. Schon bei den Übungen zur Anwendung des Karnaugh-Planes hat sich gezeigt, daß die Handhabung bei mehr als 5 Eingangsvariablen unübersichtlich und zeitaufwendig wird. Es ist leicht einzusehen, daß ein Entwurf komplexer industrieller Steuerungen bei ausschließlicher Verwendung der bisher besprochenen Grundschaltungen im allgemeinen uneffektiv ist. Die Produzenten von Automatisierungsmitteln berücksichtigen beim Entwurf von Bausteinsystemen für digitale Steuerungen diesen Sachverhalt, indem häufig benötigte Funktionsgruppen (Dekoder, Koder, Adder, Vergleicher, Register, Zähler u.ä.) zu Baueinheiten zusammengefaßt und als steckbare Leiterplatten (Module) angeboten werden. Durch die heute verfügbaren Integrationstechnologien läßt sich bereits in mittelintegrierten Schaltkreisen (MSI) der Funktionsinhalt unterbringen, der noch vor 3...5 Jahren eine Leiterplatte beanspruchte. Da der Preis eines Schaltkreises im wesentlichen von den Fertigungskosten der Masken abhängt, aber nur geringfügig durch die Anzahl der in der Schaltung benötigten Transistoren beeinflußt wird, sinkt mit steigender Integration der Preis je Gatterfunktion.

Für den Steuerungsentwickler ergibt sich damit das Problem, seine Steuerungsaufgabe möglichst vollständig mit vorgefertigten komplexen Funktionseinheiten zu realisieren. Dieses Ziel ist selbst dann zu verfolgen, wenn bei einer speziellen Steuerungsaufgabe das Vermögen eines komplexen Schaltkreises nicht völlig ausgenutzt wird. Diese Lösung ist im allgemeinen durch Wegfall von Entwurfs-, Verdrahtungs- und Dokumentationsaufwand

wesentlich besser, als eine individuell problemangepaßte Lösung mit mehreren niedrig integrierten Bausteinen.

Für die Integration von Funktionsgruppen gibt es in der Hauptsache 2 Gründe, die die Komplexität nach oben begrenzen. Einerseits ist das die Anzahl der Gehäuseanschlüsse (pin), die gegenwärtig im allgemeinen ≤ 40 realisierbar ist und nur in zwei Sonderfällen 64 beträgt;
andererseits sinkt mit Vergrößerung des Funktionsinhaltes eines Schaltkreises die Anzahl der möglichen Einsatzfälle, da umfangreichere Schaltungen im allgemeinen immer spezieller werden.
Als rentable Stückzahl eines Schaltkreises gilt heute eine Produktionshöhe von 100000 .... 1 Million Stück pro Jahr.

Zur Ausnutzung der technologischen Möglichkeiten der Größtintegration (LSI, WISI) sind deshalb teilweise neue Wege von den Bauelementeherstellern beschritten worden. Universell nutzbare Schaltungen mit einem logischen Inhalt von ≥ 1000 Gattern (außer Halbleiterspeicher) werden aus Gründen der wesentlich verbesserten Flexibilität allgemein als programmierbare Anordnungen gefertigt. Diese als Mikroprozessor, Mikroprogrammsteuerung (Microcontroler), Mikrorechner bezeichneten Strukturen besitzen die grundsätzliche Arbeitsweise der konventionellen Digitalrechner. Prinzipiell können alle Aufgaben digitaler elektronischer Steuerungen deshalb auch durch Einsatz von entsprechend programmierten Mikroprozessoren erfüllt werden. Dies ist aber nicht generell zweckmäßig. Abb. 28 zeigt, wie die Kosten einer Automatisierungslösung vom Funktionsumfang und der Anzahl gleichartiger Steuerungen abhängen.

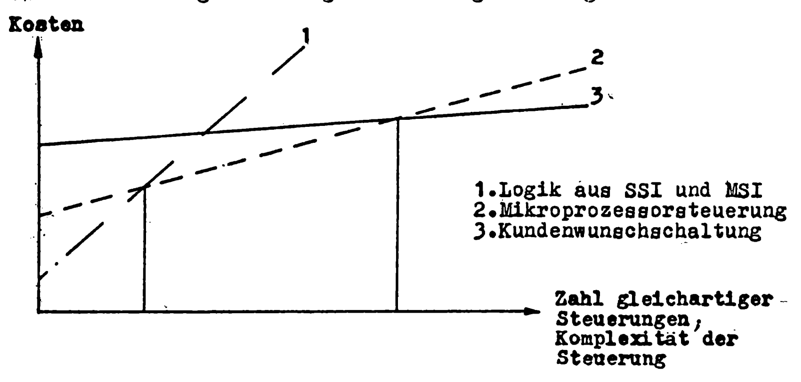



Abb. 28 Kostenfunktion verschiedener Steuerungsrealisierungen

Eine Mikrorechnersteuerung erfordert bei einfachen Problemen höhere Investitionen und für die Programmierung speziell qualifizierte Kader und im Gegensatz zu Lösungen aus Logikmodulen eine anders ablaufende Einsatzvorbereitung. Als technisches Problem tritt bei Rechnersteuerungen oft eine prozeßseitig nicht vertretbare Verarbeitungszeit (Totzeit) auf, die sich als Summe der in der Zentraleinheit des Rechners nacheinander ablaufenden Routinen ergibt. Zum genaueren Kennenlernen dieser neuartigen Technik werden in der Zukunft mit speziell zugeschnittenen Lehrbriefen Lehrgänge durchgeführt. Die Fragen in Abb. 29 sind als Entscheidungshilfe bei der Grobauswahl eines geeigneten Steuerungskonzeptes zu durchlaufen.

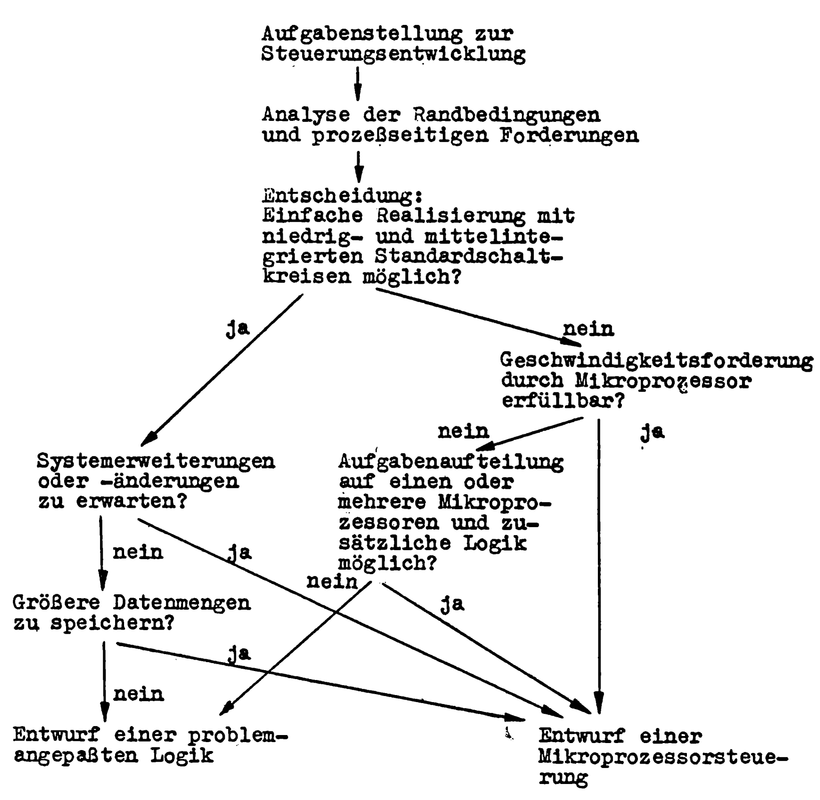



Abb. 29 Entscheidungshilfe zur Steuerungsrealisierung

Im Rahmen unserer Einführung wollen wir frei programmierbare Rechnersysteme ausklammern und uns jetzt mit den mittelintegrierten Standardschaltungen beschäftigen.

## 5.1. Speicherfreie Schaltungen

### 5.1.1. Dekoder

Eine einfache aber relativ häufige Aufgabenstellung in Steuerungen besteht darin, daß ein Stellglied oder eine Meldeleuchte nur dann aktiv werden soll, wenn eine ganz bestimmte Kombination der vorhandenen Eingangs- oder Zwischenvariablen (Zustand) auftritt. Aus den Grundlagen des Lehrbriefes 1 wissen wir schon, daß bei den 2 möglichen Zuständen einer Variablen sich für n Variable die max. Anzahl unterscheidbarer Kombinationen (k) als n-te Potenz zur Basis 2 ergibt.

$$k = 2^n \tag{1}$$

Stellen wir uns statt der einzelnen Meldeleuchte eine gasgefüllte Ziffernanzeigeröhre (Digitron) vor, dann soll bei einem Zustand die 1 bei einem anderen die 2 usw. leuchten. Wird diese Aufgabenstellung verallgemeinert, so kommen wir zur Definition eines Dekoders.

Als Dekoder werden kombinatorische Schaltungen bezeichnet, die bei m-vereinbarten Kombinationen der vorhandenen n-Eingangsvariablen m-Ausgänge besitzen und bei denen eine Logik in der Form existiert, daß bei <u>einer</u> Eingangskombination nur <u>ein</u> Ausgang ein aktives Potential führt. Ist die Bedingung m = k erfüllt, so spricht man von vollständiger Dekodierung. In Abb. 30 ist als Beispiel die Wertetabelle für die vollständige Dekodierung von 2 Eingangsvariablen gezeigt.

| B | A | $Y_3$ | $Y_2$ | $Y_1$ | $Y_0$ |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 |

Abb. 30 Wertetabelle eines $\binom{4}{1}$ Dekoders

Für m = k ergeben sich nach Formel (1) 4 Kombinationen. Da nach der Wertetabelle zu jedem Zeitpunkt nur 1 Ausgang wahr (1) ist, wird dieser Sachverhalt auch als 1- aus 4-Kode bezeichnet und teilweise in der Literatur mit den Binominalkoeffizienten $\binom{4}{1}$ abgekürzt beschrieben. Für die Realisierung von Dekodern müssen die Eingangsvariablen positiv und negiert verfügbar sein, um bei jeder gewünschten Eingangskombination je eine UND-Funktion zu erfüllen. Der Logikplan zum Beispiel aus der Abb. 30 ist in Abb. 31 dargestellt.

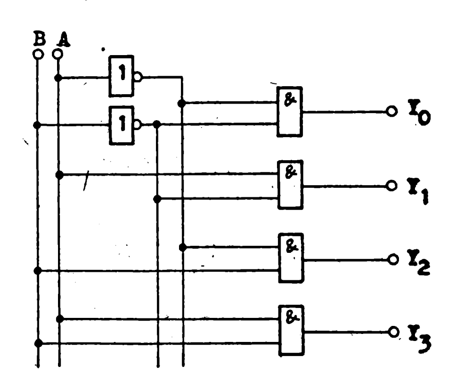



Abb. 31 Logik des Dekoders nach Abb. 30

## 5.1.1.1. Zahlensysteme, Zahlendarstellung

Die häufigste Anwendung finden Dekoder bei der Erkennung von Ziffern und Zahlen. Aus diesem Grund wollen wir hier die Grundlagen über die Zahlensysteme einschieben. Die uns überall umgebende Zahlenwelt mit den arabischen Ziffern stellt ein Positionssystem dar, d.h., der Vorrat an Ziffern beschränkt sich auf die Zeichen 0, 1, 2, 3, 4, 5, 6, 7,8, 9. Die Position einer Ziffer innerhalb einer Zahl sagt uns, ob es sich um Einer, Zehner, Hunderter usw. handelt, da der Wert (W) einer Zahl sich aus der Summe der Produkte jeder Ziffer (Z) mit dem zugehörigen Stellenwert ergibt (2).

$$W = \sum_{i=0}^{n} Z_i \cdot B^i \quad (i = 0,1,2,\ldots.n) \qquad (2)$$

B stellt dabei die Basis des Zahlensystems dar und ist stets gleich der Menge der zugelassenen Ziffernzeichen. Am Beispiel der Jahreszahl 1980 wollen wir das eben Gesagte deutlich machen. In unserer üblichen Darstellung kennen wir 10 mögliche Ziffern. Die Basis unseres Zahlensystems ist gleich 10. Wir bezeichnen es deshalb als Dezimalsystem.

$$1980 = 1 . 10^3 + 9 . 10^2 + 8 . 10^1 + 0 . 10^0$$
$$1980 = 1000 + 900 + 80 + 0$$

Die Gleichung (2) können wir aber genauso auf unsere binären Zeichen 0,1 anwenden. Da in diesem Fall nur 2 Ziffern zur Verfügung stehen, wird B = 2 und damit von einem Dualsystem gesprochen. Auf dieser Grundlage kann man jeder beliebigen Kombination mehrerer binärer Zeichen einen Zahlenwert zuordnen. Eine vereinbarte Zuordnung heißt Kode.

Beispiel:

$$10110 = 1 . 2^4 + 0 . 2^3 + 1 . 2^2 + 1 . 2^1 + 0 . 2^0$$
$$10110 = 1 . 16 + 0 . 8 + 1 . 4 + 1 . 2 + 0 . 1$$
$$10110 = 22$$

Sehen wir uns jetzt noch einmal die Abb. 30 an und ordnen der Variablen A den Stellenwert $2^0$ und B den Wert $2^1$ zu.
Man erkennt, daß die Indizes für Y nicht willkürlich gewählt sind, sondern genau dem Zahlenwert entsprechen. Für die Darstellung mehrstelliger Dezimalzahlen mittels der binären Zeichen 0,1 gibt es 2 prinzipielle Möglichkeiten, die in der Tabelle (Abb. 32) aufgeführt sind.

In der 2. Spalte ist jeder Dezimalzahl aus Spalte 1 eine Binärfolge gegenübergestellt, bei der die Summe der Produkte aus Binärzeichen und entsprechender Potenz zur Basis 2 den äquivalenten Wert darstellt. In der 3. Spalte wurde die Basis 10 aus dem Dezimalsystem beibehalten und jede dezimale Ziffer aus der Spalte 1 für sich dual kodiert. Durch diese Maßnahme stimmen die Kombinationen der Spalten 2 und 3 in den ersten 10 Zeilen überein.

| Dezimal-zahl | Dualzahl | | | | | binär kodierte Dezimalzahl (BCD) | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | $10^1$ | | | | $10^0$ | | | |
| | $2^4$ | $2^3$ | $2^2$ | $2^1$ | $2^0$ | $2^3$ | $2^2$ | $2^1$ | $2^0$ | $2^3$ | $2^2$ | $2^1$ | $2^0$ |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 2 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 3 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| 4 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 5 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |
| 6 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |
| 7 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| 8 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 9 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 10 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 11 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| 12 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| 13 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 |
| 14 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| 15 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 |
| 16 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 |
| 17 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 |
| 18 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| 19 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 |
| 20 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 21 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 |
| ⋮ | | | ⋮ | | | | | ⋮ | | | | ⋮ | |

Abb. 32 Darstellung von Zahlen im BCD und Dualkode

Außer der durch seine Gewichte innerhalb einer Tetrade als 8-4-2-1 Kode bezeichneten Form binär kodierter Dezimalzahlen gibt es noch eine Vielzahl anderer möglicher Zuordnungen von 4 Bit zu den 10 dezimalen Ziffern. Auf diese wird in diesem Rahmen nicht eingegangen /1/.

5.1.1.2. Schaltungspraxis

Dekoder wurden unter Verwendung diskreter Bauelemente meist als Diodenmatrix angeordnet. Die Abb. 33 zeigt einen $\binom{10}{1}$ Dekoder zur Entschlüsselung der 10 möglichen Ziffern einer im 8-4-2-1 Kode vorliegenden Dezimalstelle. Dabei ist zur Schaltungsvereinfachung berücksichtigt, daß von den in einer Tetrade möglichen 16 Kombinationen nur 10 praktisch zugelassen sind.

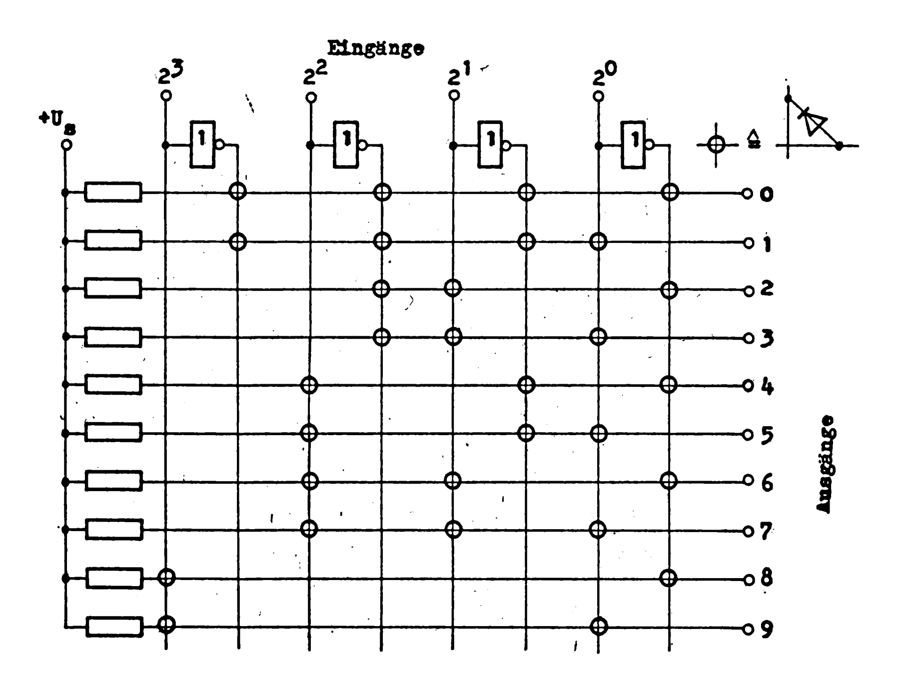



Abb. 33 **Minimierte Diodenmatrix als 8-4-2-1 Kode - Dezimal-Dekoder $\binom{10}{1}$**

**Die unvereinbarten Kombinationen (Pseudotetraden) mit einem äquivalenten Wert 10....15 ergeben Ø-Felder im Karnaugh-Plan (LB 1, Punkt 3.2). Dadurch werden statt 40 nur 30 Dioden benötigt.**

Bei Realisierungen mittels TTL-Bausteinen stehen aus DDR-Produktion nur NAND-Elemente zur Verfügung, so daß zur Einsparung von nachgeschalteten Negationen das low-Potential (L) als aktiver Zustand definiert wird. Durch Erweiterung der Eingänge zweier Negatoren mittels je 2 Dioden SAY 18 o.ä. werden insgesamt nur 4 IS benötigt. (Abb. 34).

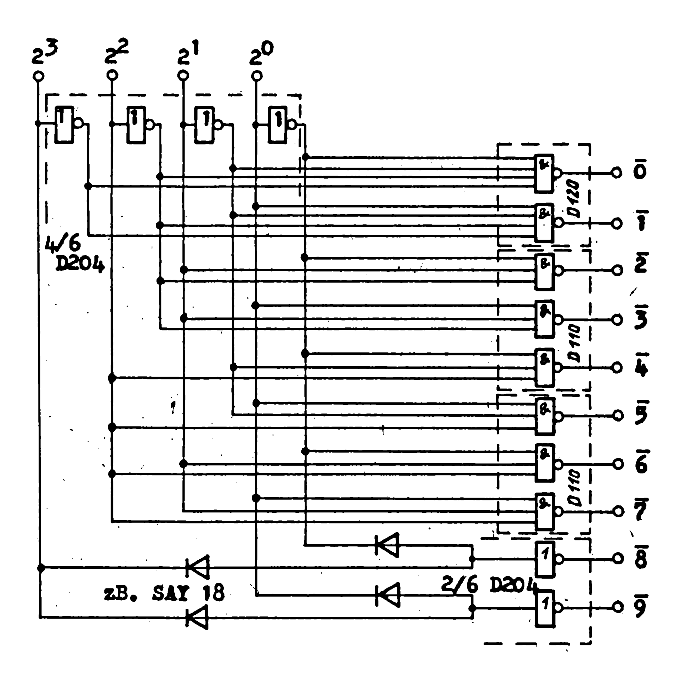



Abb. 34 8-4-2-1 Kode - Dezimal-Dekoder mit TTL-Bausteinen

Für die Dekodierung von 4 und mehr kodierten Eingängen sinkt der Schaltkreisaufwand, wenn man möglichst gleich große Gruppen ( $\leq$ 4) Eingangsvariablen bildet und diese zunächst für sich dekodiert und anschließend die entsprechenden Ausgänge der Gruppen wiederum über UND-Funktion zusammenfaßt. Bei den NAND-Elementen der TTL-Technik bedeutet das, daß in der ersten Entschlüsselungsstufe den NAND-Elementen noch Negatoren nachgeschaltet werden müssen oder entsprechend der Formel (3) die

benötigte 2stufige UND-UND-Struktur durch eine NAND-NOR-Struktur zu realisieren ist.

$$Y = (A B)(C D)$$
$$Y = \overline{\overline{A B} \vee \overline{C D}} \quad (3)$$

Besonders leicht verständlich wird die 2stufige Dekodierung am Beispiel der Zahlen 00....99, wenn diese im BCD (Abb. 32) vorliegen. Je Dezimalstelle (Tetrade) kommt eine Schaltungsanordnung nach Abb. 34 in der 1. Stufe zum Einsatz. In der 2. Stufe wird jeder Ausgang der Stelle $10^1$ jeweils mit den 10 Ausgängen der Stelle $10^0$ über NOR verknüpft. Das Prinzip ist in Abb. 35 angedeutet.

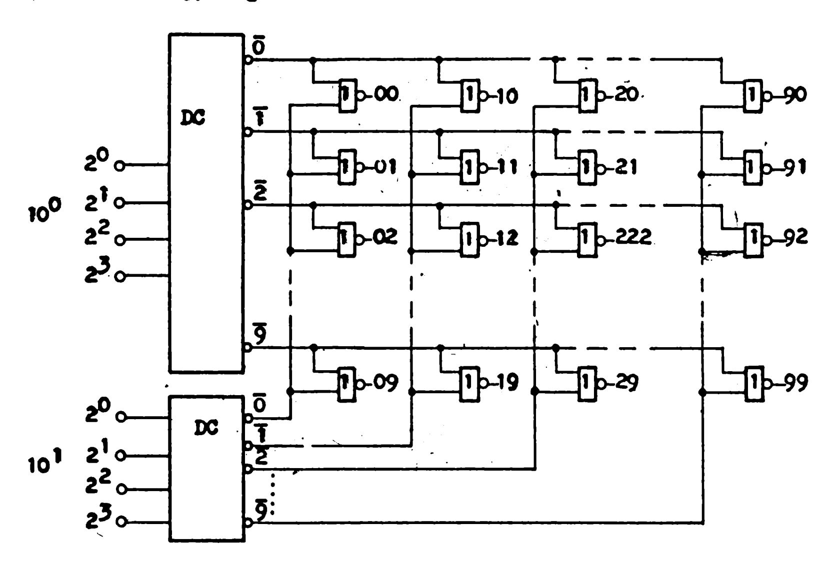



Abb. 35 2stufige Dekodierung 00....99

Als letztes Problem soll auf die möglichen Schwierigkeiten hingewiesen werden, die entstehen, wenn an die Dekoderausgänge ungetaktete Speicherelemente angeschlossen werden. Da allgemein nicht sichergestellt ist, daß alle Eingangssignale der Dekoderschaltung zum genau gleichen Zeitpunkt ihren Zustand wechseln, können im Verlauf einer Änderung der Eingangsbelegung

an mehreren Ausgängen kurzzeitige (flüchtige) Störspitzen auftreten, die u.U. als aktives Potential einen Speicher setzen und damit zu bleibenden Fehlern führen. In Abb. 36 ist dieser Sachverhalt am Beispiel des Überganges 5-6 gezeigt, wobei das Signal $2^0$ etwa 20 ns vor dem Signal $2^1$ seinen Zustand ändern soll. (Die Begründung für diesen Sachverhalt erfolgt bei den im LB 3 erklärten asynchronen Zählern.)

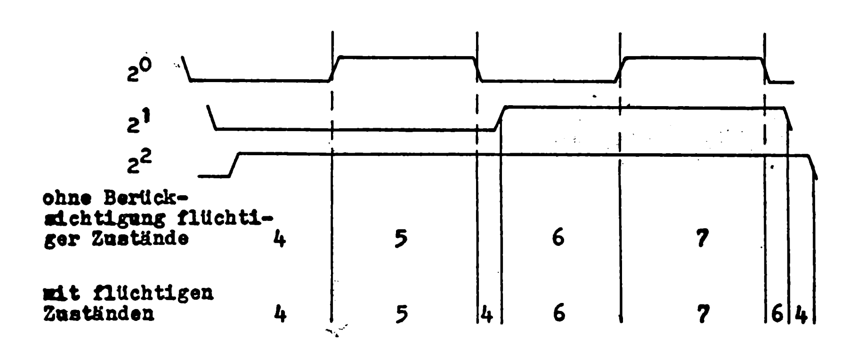



Abb. 36 Flüchtige Zustände bei Ungleichzeitigkeit des Zustandswechsels

Man umgeht diese Schwierigkeit, indem jedes UND-Element vor einem Ausgang einen zusätzlichen Eingang besitzt. Diese sind alle untereinander verbunden und bilden den Freigabeeingang. Dieser auch als Strobe oder Enable bezeichnete Eingang kann durch inaktives Potential alle Ausgänge während des Zustandswechsels passiv halten und hat damit eine Takt- oder Synchronisationsfunktion.

International stehen Bausteine in integrierter Form bereit, die eine vollständige Dekodierung von 3 Eingängen ($\binom{8}{1}$) (SN 74138) oder 4 Eingängen ($\binom{16}{1}$) (SN 74154) bzw. eine Dekodierung verschiedener Dezimalkodes ($\binom{10}{1}$) (SN 7442, SN 7443) vornehmen.

Außer zur Anzeige von Zahlen mittels Ziffernanzeigeröhren werden die Dekoder als wesentlicher Bestandteil von Multiplexern und Demultiplexern (Abschnitt 5.1.4., 5.1.5.) und auch allen anderen Formen von Adressenentschlüsselungen in Rechnersystemen benötigt.

### 5.1.2. Koder

Auf der Eingangsseite von informationsverarbeitenden Einrichtungen tritt häufig das Problem der Zahleneingabe auf. Gerätetechnisch wird diese Aufgabe beispielsweise durch eine Anordnung aus 10 Tasten (z.B. Taschenrechner) oder durch einen Wahlschalter mit 10 Stellungen (Ziffernvorwahlschalter) gelöst.

Bei Betätigung einer der Tasten soll am Eingang der informationsverarbeitenden Schaltung die entsprechende Binärkombination eines vereinbarten Kodes bereitgestellt werden, weil erst durch geeignete Kodes eine maschinelle, rechentechnische Behandlung der eingegebenen Zahlen (Abschnitt 5.1.7.) möglich wird. Es gibt noch einen zweiten Grund zur Kodierung, der auch für beliebige Zeichenmengen zutrifft.

Das lateinische Alphabet umfaßt 26 Buchstaben. Würde ohne Kodierung ein Fernschreiber konstruiert, so müßten zwischen den 26 Tasten des Senders und den 26 Betätigungsmagneten des Empfängers 26 Signalleitungen verlegt werden. Durch Kodierung läßt sich dieser Aufwand wesentlich vermindern. Durch Umkehrung der Gleichung (1) kann man errechnen, wieviel binäre Variable n mindestens benötigt werden, um m unterscheidbare Zustände zu kodieren.

$$n \geq \log_2 m \qquad (n = \text{ganzzahlig}) \tag{4}$$

$$n \geq \operatorname{ld} m$$

$$\operatorname{ld} m = \operatorname{ld} 10 \cdot \lg m \tag{5}$$

$$\operatorname{ld} m \approx 3{,}32 \cdot \lg m$$

$$n \geq 3{,}32 \cdot \lg m \tag{6}$$

Für unser vereinfachtes Beispiel eines Fernschreibers erhalten wir durch Einsetzen

$$n \geq 3{,}32 \quad \lg 26$$
$$n \geq 3{,}32 \quad 1{,}415$$
$$n = 5$$

Wird ein Kode vereinbart, bei dem n größer ist als der Wert aus Gleichung (6), so sind Redundanzen vorhanden, die beispielsweise zur Übertragungssicherung genutzt werden können.

Wenden wir uns nach der kurzen theoretischen Betrachtung wieder praktischen Problemen zu. Für das Beispiel der Zehner-Tastatur stellt die Abb. 32 die Wertetabelle dar, wenn wir nur die Zeilen 0 ..... 9 ansehen. Für die Ausgangsvariable $2^0$ sehen wir sofort, daß diese 1 werden soll, wenn die Zifferntaste 1 ODER 3 ODER 5 ODER 7 ODER 9 aktiviert wird. Da sich für die anderen Ausgangsgrößen ein analoger logischer Zusammenhang ergibt, merken wir uns:

**Koder sind als kombinatorische Schaltung realisierbar, die unter der Bedingung, daß zu einem Zeitpunkt nur einer der m-Eingänge aktiv ist $\binom{m}{1}$, mit einem ODER-Glied je Ausgang arbeitet.**

In der diskreten Technik wurden die benötigten ODER wiederum als Diodenmatrix angeordnet. In Abb. 37 ist die Schaltung zur Generierung des 8-4-2-1 Kodes aus den $\binom{10}{1}$ Signalen der dezimalen Ziffern zu sehen.

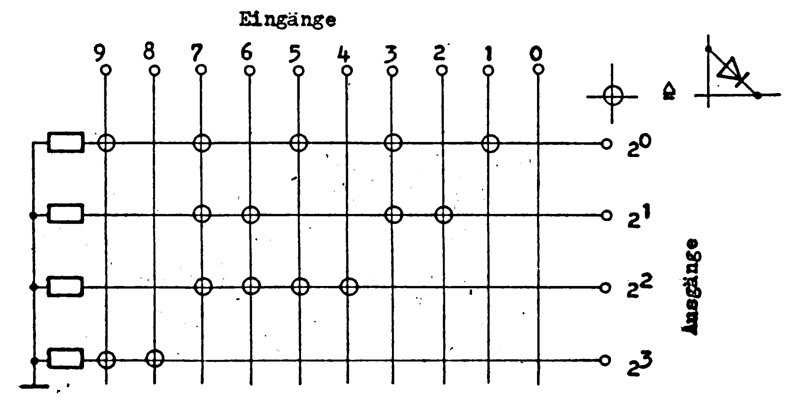



Abb. 37 8-4-2-1 Koder als ODER-Diodenmatrix

Für die Realisierung der ODER-Funktion in Kodern kann man die NAND-Gatter der TTL-Technik einsetzen, wenn man die Eingangssignale negiert bereitstellt.

$$A \vee B = \overline{\overline{A \vee B}} = \overline{\bar{A}\,\bar{B}} \qquad (7)$$

Die Schaltung weist keine Besonderheiten auf.

Etwas modifiziert ist die Kodierung bei Ziffernvorwahlschaltern (Abb. 38) gelöst. Die zur Stellungswahl drehbar gelagerte Ziffernscheibe trägt 4 Schleifbahnen im 8-4-2-1 Kode, so daß direkt an 4 Ausgängen die Kodierung abgenommen werden kann.

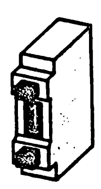


Abb. 38 Ansicht eines Ziffernvorwahlschalters

Zum Aufruf besitzt jede Dekade eine gemeinsame Eingangsleitung (Abb. 39).

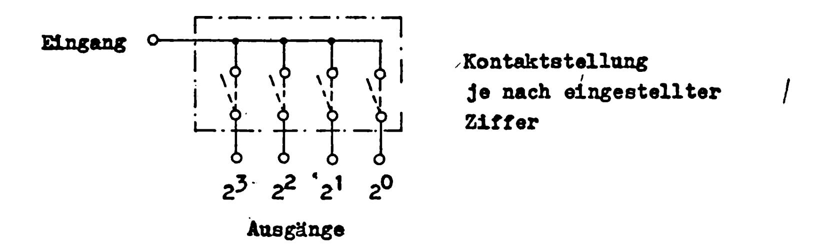



Abb. 39 Innenschaltung eines Ziffernvorwahlschalters (kodiert)

Diese Form von Kodern bringt Vorteile in Steuerungen, wenn an einer Einheit (voreinstellbarer Zähler oder Vergleicher) zwischen mehreren manuell einstellbaren Werten je nach Betriebszustand automatisch einer ausgewählt werden soll. Die Abb. 40 zeigt die Schaltung für 2 Werte.

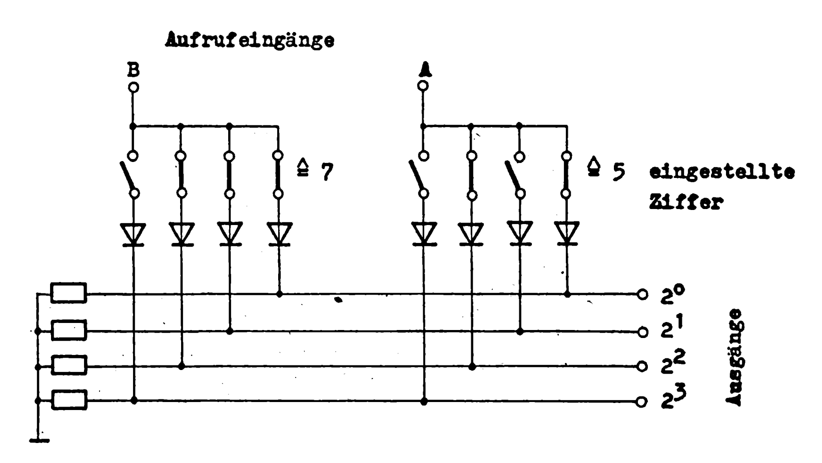



Abb. 40 Ziffernauswahlschaltung

Zur Vereinfachung ist nur eine Dekade gezeichnet. Durch die Signale A bzw. B wird der Wert 5 bzw. 7 aufgerufen. Die Dioden entkoppeln die Schalter untereinander und realisieren eine ODER-Funktion, die in Verbindung mit dem Aufruf die Betriebsweise eines Multiplexers nachbildet. Im Abschnitt 5.1.4. wird für Multiplexer auch eine einfache Realisierung mit dem TTL-Baustein D 103 (offener Kollektor) angegeben.

### 5.1.3. Umkodierer

Bei der Kopplung von 2 digitalen Baueinheiten ergibt sich teilweise das Problem, daß eine Zeichenmenge aus einem Kode in einen anderen überführt werden muß. Ein naheliegendes Beispiel ist für uns die Anzeige des 8-4-2-1 Kodes mit 7 - Segment - anzeigen (siehe entsprechende Übungsaufgabe zu Kapitel 3 des

Lehrbriefes 1), die uns aber nicht weiter beschäftigen soll, da ein industrieller mittelintegrierter Baustein (D 147) für diese Aufgabe existiert. Während in diesem Fall die 7 Ausgangsvariablen (Segmente der Anzeige, z.B. VQB 71) jeweils eine Bool'sche Funktion von nur 4 Eingangsvariablen (duale Stellenwertigkeiten $2^3$, $2^2$, $2^1$, $2^0$) darstellen, gibt es in der Praxis noch wesentlich komplexere Aufgabenstellungen. Dazu gehört beispielsweise die Umkodierung der ≈60 Zeichen (Buchstaben, Ziffern und Sonderzeichen) des EIA-8-Bit-Kodes in den ISO-8-Bit-Kode oder die Umkodierung dual verschlüsselter Ziffern 00....99 in die BCD-Darstellung (siehe Tabelle Abb. 32).

Unter Umkodierung wollen wir stets Funktionenbündel verstehen. Eine sehr übersichtliche und leicht zu entwerfende Schaltung ergibt sich, wenn man zunächst die n-Eingangsvariablen in m-Zeichen und diese als Zwischenvariablen vorliegenden Größen nachfolgend neu in die l-Ausgangsgrößen kodiert. Diese einfache Vorschrift führt immer zum Erfolg. Sie garantiert aber keine aufwandsminimale Realisierung. Bei den modernen Bauelementen der Mikrorechentechnik ist diese Vorgehensweise jedoch besonders effektiv. In einem anwenderprogrammierbaren Lesespeicher (PROM) wird durch jede Eingangskombination ein Speicherplatz adressiert. Auf diesen schreibt (programmiert) der Anwender die gewünschte Ausgangskombination. Damit entsteht ein Umkodierer (Zuordner), der rein kombinatorisch arbeitet. Solche Bausteine sind heute schon mit 10 Eingängen, 1024 Plätzen und je Platz 8 Ausgangsvariablen verfügbar. Aber auch ohne Benutzung von PROM-Elementen ist diese Entwurfsstrategie stets vorteilhaft, wenn zumindest für die Dekodierung angebotene mittelintegrierte Bausteine eingesetzt werden können.

Eine 2. Methode für den Entwurf von Umkodierern ergibt sich aus der direkten Funktionsaufstellung aus der Schaltbelegungstabelle.

An dem nachfolgenden Beispiel wollen wir beide Verfahren demonstrieren.

Am Ausgang einer numerischen Steuerung werden die zur Drehbearbeitung benötigten Hauptspindeldrehzahlen durch 8 potentialfreie Relaiskontakte in BCD-Darstellung bereitgestellt. Die

angeschlossene Drehmaschine besitzt in unserem Beispiel einen Hauptspindelmotor mit konstanter Drehzahl und ein über 5 Magnetventile schaltbares Getriebe, das ausgangsseitig die 6 in Abb. 41 angegebenen Drehzahlen realisiert.

| Drehzahl Dezimal | (Eingänge) BCD-Darstellung $10^1$ | | $10^0$ | Magnetventile (Ausgänge | | | | |
|---|---|---|---|---|---|---|---|---|
| | | H G F E | D C B A | Y5 | Y4 | Y3 | Y2 | Y1 |
| 0 | | 0 0 0 0 | 0 0 0 0 | 0 | Ø | Ø | Ø | Ø |
| 10 | | 0 0 0 1 | 0 0 0 0 | 1 | 0 | 0 | 1 | 1 |
| 16 | | 0 0 0 1 | 0 1 1 0 | 1 | 0 | 1 | 0 | 1 |
| 25 | | 0 0 1 0 | 0 1 0 1 | 1 | 0 | 1 | 1 | 0 |
| 40 | | 0 1 0 0 | 0 0 0 0 | 1 | 1 | 0 | 1 | 0 |
| 63 | | 0 1 1 0 | 0 0 1 1 | 1 | 1 | 1 | 0 | 0 |

Abb. 41 Schaltbelegungstabelle eines Umkodierers (Beispiel)

Beim Entwurf der Umkodierschaltung ist zu beachten, daß bei der aufgabengemäßen Zusammenschaltung von Steuerung und Maschine nur die 6 vereinbarten BCD-Kombinationen eingangsseitig auftreten können.

Als erste Lösung wollen wir die Funktion für die Magnetventile direkt aus der Schaltbelegungstabelle aufstellen. Ohne Anwendung von Vereinfachungsregeln ergibt sich für das Magnetventil Y1

$$Y1 = \overline{H}\,\overline{G}\,\overline{F}\,E\,\overline{D}\,\overline{C}\,\overline{B}\,\overline{A} \vee \overline{H}\,\overline{G}\,\overline{F}\,E\,\overline{D}\,C\,B\,\overline{A}$$

Auch für die MagnetventileY2 ..... Y5 ergeben sich ähnlich umfangreiche Ausdrücke. Eine Minimierung mittels Karnaugh-Plan erscheint bei 8 Eingangsvariablen im ersten Moment unmöglich. Bei genauerer Betrachtung fällt aber auf, daß H bzw. D entfallen, da sie konstant 0 sind. Weiterhin ist zu erkennen, daß die Dezimale $10^1$ bis auf die Unterscheidung der Drehzahlen 10 und 16 zur eindeutigen Beschreibung ausreicht. Für die Differenzierung von 10 und 16 müssen wir aus der Dezimale $10^0$ noch eine Variable hinzunehmen, die bei 10 und 16 unterschiedlich belegt ist. Geeignet ist sowohl C als auch B. Wir wählen C.

Unsere Karnaugh-Pläne für die 5 Ausgangsgrößen enthalten jetzt nur noch die Variablen G, F, E, C und dienen nach den bekannten Regeln zur Vereinigung der Ausdrücke.

Für das Magnetventil Y3 ist die Ermittlung des logischen Ausdruckes (Abb. 42) zur Wiederholung angegeben.

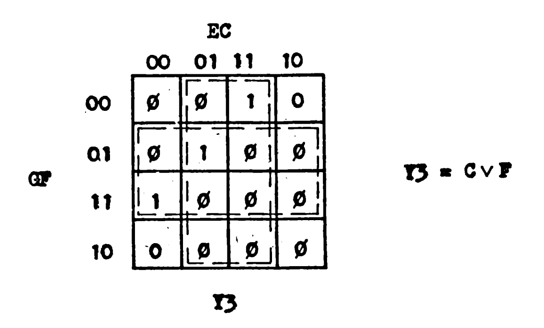



Abb. 42 Karnaugh-Plan für Magnetventil Y3

Insgesamt ergeben sich folgende Gleichungen:

Y1 = E

Y2 = $G.\bar{F}$ c $E\,\bar{C}$ v $\bar{E}$ C

Y3 = F v C

Y4 = G

Y5 = G v E v C

In Abb. 43 ist die vollständige Schaltung mit TTL-Bausteinen dargestellt.

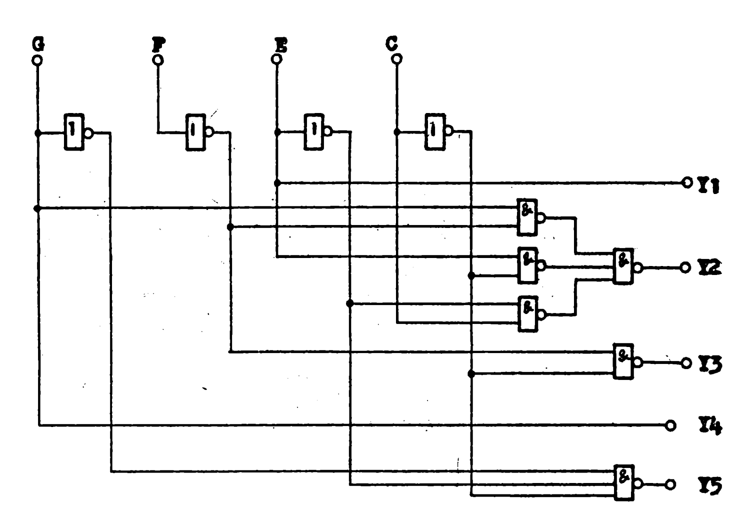



Abb. 43 Umkodierer aus NAND-Gattern bei direkter Funktionsrealisierung (Beispiel)

Wenden wir uns nun der Realisierung des Umkodierers mit mittelintegrierten Dekoderbausteinen zu. Es soll der im Abschnitt 5.1.1.2. erwähnte Schaltkreis SN 74138 zum Einsatz kommen. Mit ihm dekodieren wir die Tetrade $10^1$. Um 10 und 16 zu unterscheiden, wird der Ausgang 1, wie in Abb. 44 gezeigt, durch die Variable aufgesplittet .

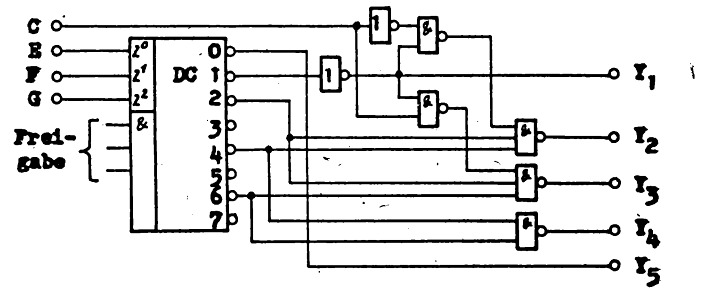



Abb. 44 Umkodierer mit mittelintegriertem Dekoder SN 74138 (Beispiel)

Zu beachten ist, daß alle Ausgänge des Dekoders low-aktiv sind.

Bei Einsatz des $\binom{16}{1}$ Dekoders SN 74154 würde die Logik zur Trennung von 10 und 16 vom Schaltkreis mit übernommen.

### 5.1.4. Multiplexer

In der kontaktbehafteten Schaltungstechnik wird mit Wechslern von Relais bzw. ein- und mehrpoligen Umschaltern häufig die Aufgabe gelöst, Signale bzw. Signalgruppen von 2 oder mehreren Quellen auf einen Kanal wahlweise aufzuschalten. In der digitalen Elektronik werden solche Einheiten als Multiplexer bezeichnet.

Der einfachste Fall ist ein 2- zu-1-Multiplexer. Das Steuersignal (S), das bei Relaisausführung der Spulenspannung entsprach und damit festlegte, ob der Arbeits- oder Ruhekontakt geschlossen war, wird auch in der kontaktlosen Technik benötigt. Es wird im Multiplexer als wahrer und negierter Wert bereitgestellt und je einem UND-Tor zugeführt. Die Ausgänge der UND-Tore werden nachfolgend zu einem Ausgang ODER verknüpft. Während bei der Relaisvariante die Anzahl der Wechsler je Relais begrenzt ist (meist 2, höchstens 6) und beim Parallelbetrieb von Relais die Umschaltzeitpunkte stark unterschiedlich werden können, kann man beliebig viele Multiplexer problemlos parallel betreiben.
In Abb. 45 ist ein 2-zu-1-Multiplexer für 4-Bit-breite Werte (Tetraden) angegeben, der aus 2 Schaltkreisen D 150 aufgebaut ist (negierte Ausgänge).

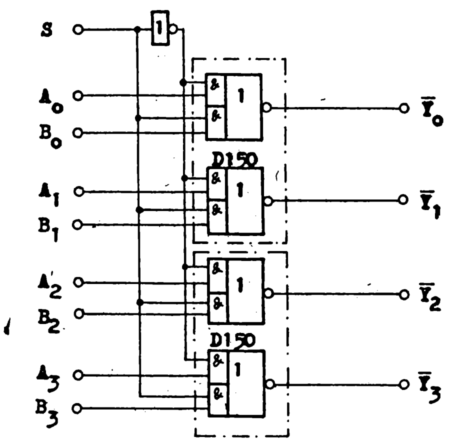



Abb. 45 2-zu-1-Multiplexer aus 2 x D 150 für 4-Bit-breite Werte

In integrierter Form wird ein Schaltkreis mit diesem Funktionsumfang international als SN 74157 gefertigt.

Bei der Umschaltung von mehr als 2 Eingangskanälen auf einen Ausgangskanal gibt es manuell betätigt mehrstellige Umschalter (beispielsweise als Betriebsartenwahlschalter). Mit Relais ergeben sich Reihenschaltungen von Wechslern in einer "Baumstruktur". In der integrierten Lösung ist bei einem k-zu-1-Multiplexer ein $\binom{k}{1}$ Dekoder vorhanden (Verringerung der Steuereingänge), wobei dessen Ausgänge k UND-Tore steuern, die über ein ODER zusammengefaßt sind.

<u>Ein Multiplexer besteht aus disjunktiv verknüpften Torschaltungen, die durch $\binom{k}{1}$ Signale geöffnet werden.</u>

In integrierter Form werden heute schon 16-zu-1-Multiplexer (SN 74150 bzw. K 155 KP 1) angeboten, wobei die Anzahl der umschaltbaren Kanäle nur durch die Anzahl der Anschlüsse begrenzt ist. An dieser Stelle sei noch ein Anwendungsbeispiel für einen 4-zu-1-Multiplexer angeführt. In einer Recheneinrichtung (Abschnitt 5.1.7.) wird in Abhängigkeit von 4 Betriebszuständen (in $S_0$, $S_1$ kodiert) am Eingang eines 4-Bit-parallelen Adders entweder der wahre Wert einer Meßgröße A ($A_3$, $A_2$, $A_1$, $A_0$) oder deren negierter Wert A ($\bar{A}_3\bar{A}_2\bar{A}_1\bar{A}_0$) oder die Konstante 15 (1 1 1 1) oder die Konstante 0 (0 0 0 0) benötigt. Die Aufgabe läßt sich mit 2 integrierten Schaltungen SN 74153 bzw. K 155 KP 2 lösen. Aus Gründen der besseren Verfügbarkeit ist in Abb. 46 die Lösung mit 4 x D 154, 1 x D 100 und 1 x D 204 angegeben.

Abschließend sei noch auf 2 Sonderformen von Multiplexern verwiesen. Soll von 3 ... 5 bzw. mehr Quellen auf einen Empfänger umgeschaltet werden können, so ist der D 154 nicht voll ausgelastet oder nicht ausreichend. Es bietet sich dann eine Lösung mit den Schaltkreisen D 103 bzw. D 126 an, die als Ausgang einen offenen Kollektor aufweisen und deshalb parallel geschaltet werden dürfen. Allerdings wird für die zusammengefaßten Ausgänge ein externer Widerstand entsprechend der Tabelle in /2/ benötigt. Diese Parallelschaltung stellt ein "verdrahtetes"

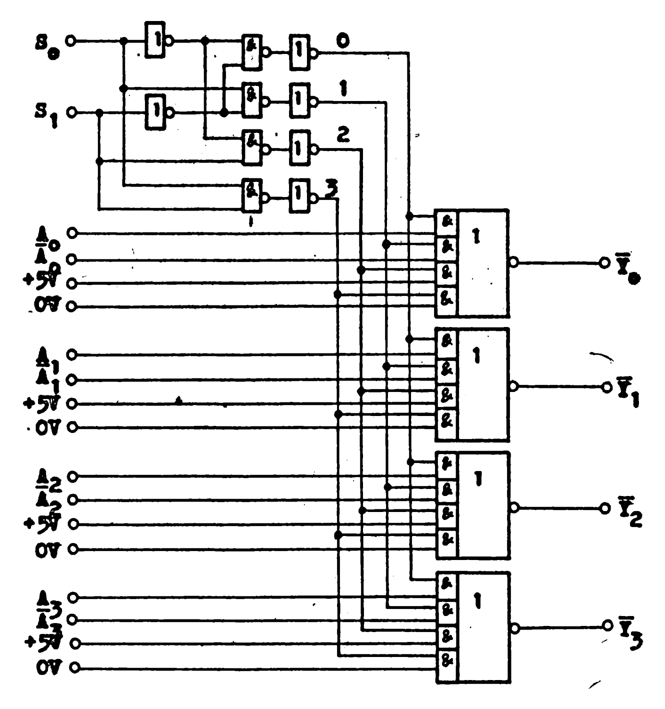



Abb. 46 Schaltung eines 4-zu-1-Multiplexers

NOR dar, da das low-Potential ausgangsseitig dominiert, d.h., wenn an einem oder mehreren parallel geschalteten NAND-Elementen die beiden Eingänge H sind, wird der jeweilige Ausgangstransistor leitend = L. Durch Negation der NOR-Funktion erhält man die für Multiplexer typische Folge: UND-Tore - ODER.

In Abb. 47 ist gezeigt, wie ein 4-Bit-breites Parallelregister aus 2 D 174 durch 3 D 103 zusätzlich zu einem aufrufbaren Links-Rechts-Schieberegister erweitert werden kann. Die Negation nach dem verdrahteten NOR kann in diesem Fall weggelassen werden, wenn man die negierten FF-Ausgänge benutzt.

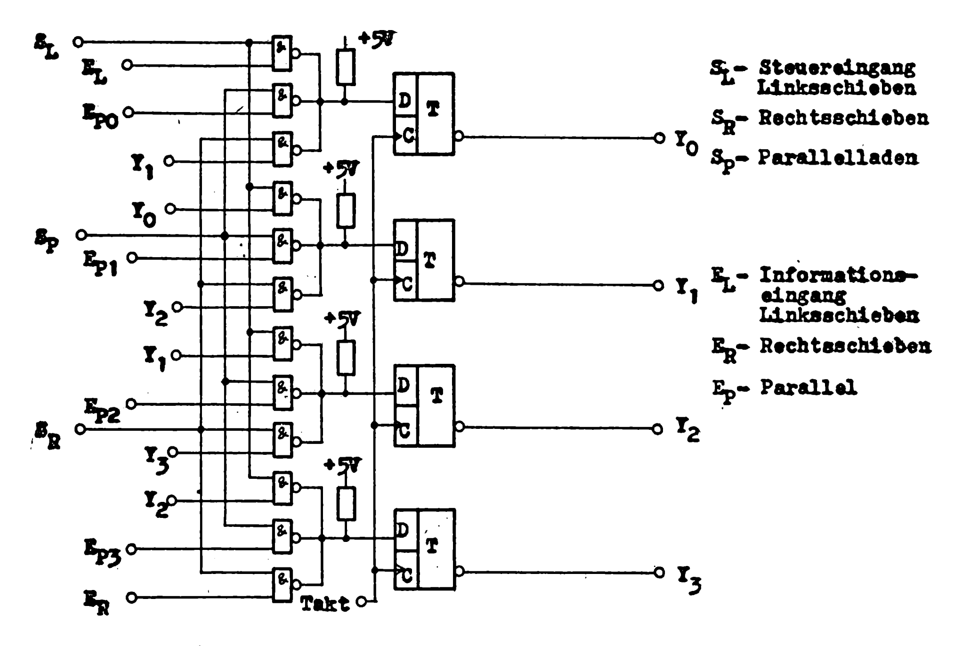



Abb. 47 3-zu-1-Multiplexer mit D 103

Die Anwendung der "verdrahteten" ODER-Funktion als Multiplexer bringt ganz entscheidende Vorteile, wenn die Quellen räumlich weit voneinander entfernt sind. Jede Quellenleitung wird mit einem offenen Kollektor ausgestattet und damit der Multiplexer dezentralisiert, so daß statt vieler Leitungen zum konventionellen Multiplexer nur jeweils eine Sammelleitung (Bus) geführt wird. Für die Rechentechnik, die häufig solche Bussysteme benutzt, wurden spezielle IS mit hochbelastbarem Gegentaktausgang geschaffen. Durch einen Steuereingang werden alle Ausgänge unabhängig von der Eingangsbelegung hochohmig (Tristate-Ausgang), so daß die Ausgänge parallel betrieben werden dürfen, wenn zu jedem Zeitpunkt nur ein einziger Steuereingang aktiv ist. Diese speziellen IS (z.B. 8216) erfüllen deshalb in Verbindung mit vorgeschalteten Dekodern auch die Funktion von Multiplexern.

## 5.1.5. Demultiplexer

Während bei kontaktbehafteten Umschaltern das bewegliche Schaltstück sowohl Ein- als auch Ausgang für ein Signal sein kann, ist ein elektronischer Multiplexer nicht "rückwärts" betreibbar. Zur Umkehrung der Funktion des Multiplexers - die wahlweise Umschaltung des Signals einer Quelle auf mehrere mögliche Empfänger - ist eine spezielle Baueinheit erforderlich, die als Demultiplexer bezeichnet wird.

Bei genauer Betrachtung erkennt man, daß die Schaltung eines Demultiplexers ein speziell ausgestatteter Dekoder ist und nur etwas anders beschaltet wird.

In integrierter Form werden die intern als $\binom{k}{1}$ benötigten Steuersignale wiederum durch einen $\binom{k}{1}$ Dekoder bereitgestellt (Adressenentschlüsselung), da durch die extern dual verschlüsselten Steuersignale Schaltkreisanschlüsse eingespart werden. Das zu einem Zeitpunkt aktive Steuersignal öffnet das gewünschte UND-Tor und gibt damit einen Ausgang frei. Das zu verteilende Quellensignal liegt an jedem vorhandenen UND-Tor parallel an.

Da sowohl Multiplexer als auch Demultiplexer mit vorgeschalteten Dekodern arbeiten, gilt zur Vermeidung kurzzeitiger Störungen während des Wechselns der Steuerbelegung (Adresse) das im Abschnitt 5.1.1. Gesagte. Der Freigabeeingang (Strobe oder Enable) bei einigen integrierten Dekodern wird auch an alle Ausgangstore geführt. Damit kann jeder so ausgestattete Dekoder als Demultiplexer genutzt werden, wenn der Enableeingang mit dem Quellensignal belegt wird. Für einen solchen IS (SN 74155)ergibt sich folgende Wertetabelle: /3/

| | Adressen | | Ausgänge | | | |
|---|---|---|---|---|---|---|
| Strobe | B | A | $Y_3$ | $Y_2$ | $Y_1$ | $Y_0$ |
| L | L | L | H | H | H | L |
| L | L | H | H | H | L | H |
| L | H | L | H | L | H | H |
| L | H | H | L | H | H | H |
| H | X | X | H | H | H | H |

X = beliebige Belegung.

Wie zu sehen ist, besitzt der angegebene Dekoder low-aktive Ausgänge. In Abb. 48 ist der zugehörige Logikplan dargestellt.

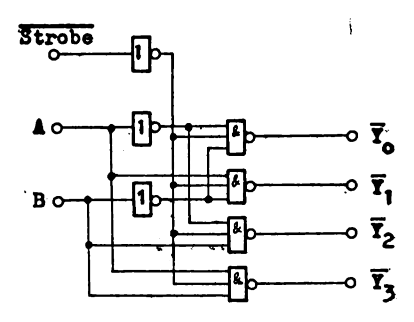



Abb. 48 Schaltung eines Dekoders mit Strobeeingang als Demultiplexer (Strobe = Dateneingang)

## 5.1.6. Vergleicher

In digitale Steuerungen werden in neuerer Zeit immer häufiger auch digitale Meßwertgeber für Lage, Spannung u.ä. einbezogen. Es besteht dann die Aufgabe, beim Erreichen von vorgegebenen, einstellbaren Sollwerten Schalthandlungen auszuführen. In der analogen Schaltungstechnik setzt man zu diesem Zweck speziell ausgelegte Operationsverstärker ein, die als Komparator bezeichnet werden. Für die in diesem Punkt behandelten Vergleicher findet man in der Literatur deshalb auch den Begriff digitaler Komparator.

Wenn nur die Gleichheit (Übereinstimmung, Koinzidenz) zweier Werte

$A\ (A_{n-1},\ A_{n-2}\ \ldots\ A_1,\ A_0),\ B\ (B_{n-1},\ B_{n-2}\ \ldots B_1,\ B_0)$

ausgewertet werden soll, reicht eine ausgangsseitig UND-verknüpfte Anordnung von n Äquivalenzschaltungen (logisches Symbol $\sim$), wenn die zu überwachenden Werte n-Bit breit sind.

$$(A = B) = A_{n-1} \sim B_{n-1} \wedge A_{n-2} \sim B_{n-2} \wedge \ldots A_1 \sim B_1 \wedge A_0 \sim B_0 \qquad (8)$$

Besonders bei Positionieraufgaben interessiert für die automatische Richtungserkennung beim Einschalten des Antriebes außer der Koinzidenz auch noch die Aussage $A>B$ oder $A<B$, so daß in integrierten Vergleichern (SN 7485) alle 3 Signale generiert werden. Da in einem Schaltkreis durch Beschränkung der Anschlußstiftzahl nur 4 Bit untergebracht werden können, ist noch je ein Kaskadeneingang ( $A^*>B^*$, $A^* = B^*$, $A^* < B^*$) für höherwertige Komparatorschaltkreise vorhanden.

Bereits für 3 zu vergleichende Bitstellen ergibt sich ein recht umfangreicher Ausdruck (9)

$$A>B = A^* > B^* \vee A_2>B_2 \vee \left[(A_2 \sim B_2)(A_1>B_1)\right] \vee \left[(A_2 \sim B_2)(A_1 \sim B_1)(A_0>B_0)\right] \quad (9)$$

Nur bei höchsten Geschwindigkeitsforderungen wird die Logik in 2 Stufen realisiert. Allgemein wird zur Verringerung des Aufwandes mit Zwischenvariablen gearbeitet (10) und (11)

$$A>B = A^*>B^* \vee A_2 \bar{B}_2 \vee (A_2 B_2 \vee \bar{A}_2 \bar{B}_2) \left[A_1 \bar{B}_1 \vee (A_1 B_1 \vee \bar{A}_1 \bar{B}_1) A_0 \bar{B}_0\right] \quad (10)$$

$$A<B = A^*<B^* \vee \bar{A}_2 B_2 \vee (A_2 B_2 \vee \bar{A}_2 \bar{B}_2) \left[\bar{A}_1 B_1 \vee (A_1 B_1 \vee \bar{A}_1 \bar{B}_1) \bar{A}_0 B_0\right] \quad (11)$$

Von den Gleichungen (8), (10), (11) brauchen insgesamt jedoch nur 2 realisiert zu werden. Da die 3 Aussagen alle möglichen Kombinationen der zu vergleichenden Werte beschreiben, ist der 3. Fall stets dann erfüllt, wenn die 2 realisierten Funktionen nicht erfüllt sind (12).

$$(A = B) = \overline{A>B} \wedge \overline{A<B} \quad (12)$$

In Abb. 49 ist ein kaskadierbarer Vergleicher für 3 Bit gezeigt, der die Formeln (10), (11) und (12) unter alleiniger Verwendung von NAND realisiert. Es fällt leicht, die angegebenen Gleichungen für beliebig breite Werte zu erweitern.

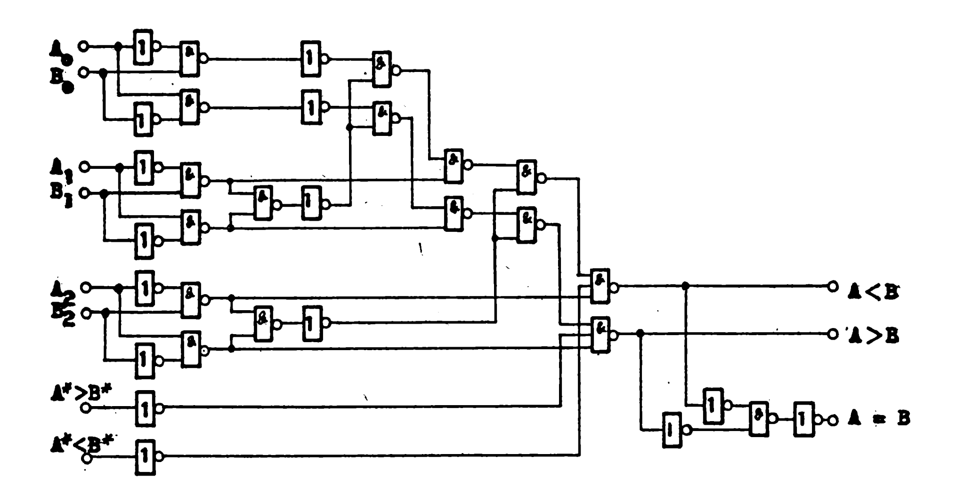



Abb. 49 Vergleicher mit niedrig integrierten Schaltkreisen

Die behandelten Vergleicher arbeiten völlig ungetaktet (asynchron). Da für sehr breite Worte (> 8 Bit) der Aufwand ohne mittelintegrierte Bausteine stark ansteigt, sind getaktet arbeitende Lösungen einfacher. Sie benötigen 2 Schieberegister (Lehrbrief 3), die zum Ring geschaltet sind, und einen Vergleicher für eine Bitstelle (Abb. 50). Nach dem parallelen Laden der beiden Register mit den zu vergleichenden Werten (12 Bit) werden die Werte einmal im Ring verschoben. Der Vergleicher erhält, mit der höchstwertigen Stelle beginnend, nacheinander (sequentiell) alle zu vergleichenden Bit. Die beiden D-FF am Vergleichereingang wurden mit dem Ladetakt der Register gesetzt. Die D-FF werden synchron mit dem Schiebetakt gesteuert bis einmal A>B oder A<B erfüllt ist. In diesem Moment wird der Takt gesperrt.

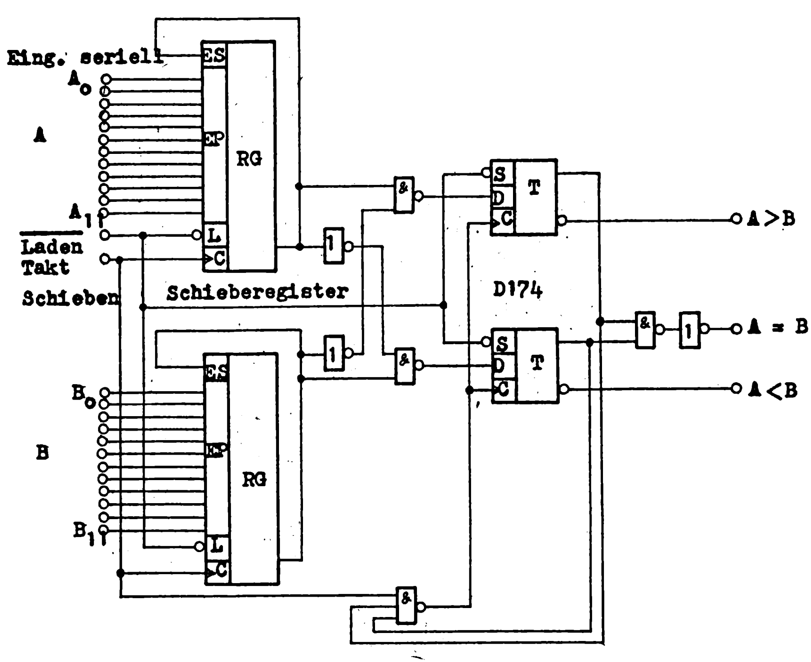



Abb. 50 Serieller Vergleicher (12 Bit)

Abschließend wollen wir noch einen besonders einfachen Fall eines Vergleichers betrachten, wenn nur die Gleichheit interessiert. Er setzt voraus, daß der Ist-Wert nur von einer Seite (nicht alternierend) auf den ansonsten beliebigen Soll-Wert einläuft. Diese Bedingung ist beispielsweise bei allen Zählvorgängen erfüllt.

Wird der größere der beiden Werte negiert bereitgestellt, so muß statt eines Äquivalenzgliedes pro Bitstelle nur ein ODER eingesetzt werden. Die ODER-Ausgänge werden wieder über ein UND zusammengefaßt.

Als Beispiel wollen wir annehmen, daß an einem Ziffernvorwahlschalter als Soll-Wert die Stückzahl einer Ware eingestellt wird, die pro Verpackungseinheit portioniert werden soll. Der Ist-Wert wird durch eine Lichtschranke und einen nachgeschalteten Zähler (Lehrbrief 3) erfaßt. Der Zähler wird vor jeder

Zählung auf Null gestellt. In Abb. 51 ist die Schaltung mit NAND-Gattern für eine Dekade ausgeführt.

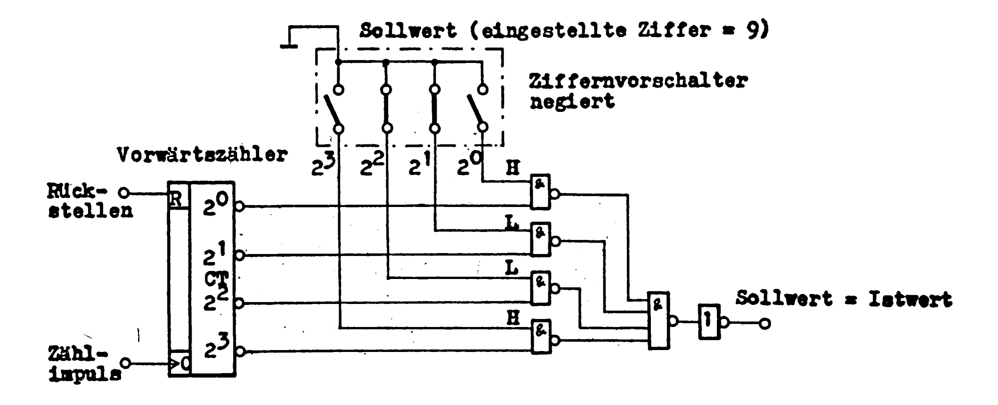



Abb. 51 Koinzidenzschaltung für homogen wachsende Ist-Werte

Dabei ist zu beachten, daß für die geforderte ODER-Funktion negierte Eingangsdaten am NAND benötigt werden, d.h., daß der Soll-Wert als größerer Wert unnegiert angelegt werden muß. Dies erreicht man nur mit negiert kodierten Ziffernvorwahlschaltern, die am Aufrufeingang mit low-Potential beaufschlagt werden, weil ein offener TTL-Eingang high-Potential darstellt.

## 5.1.7. Adder

Im Abschnitt 5.1.6. haben wir für Positionieraufgaben Vergleicher eingesetzt, die an 3 Ausgängen in dekodierter Form ($\frac{3}{1}$) die Information liefern, ob A>B oder A<B ist und damit die Drehrichtung eines Antriebes so steuern können, damit die beiden Werte durch die erfolgende Veränderung gleich groß werden (A=B). Wird mit großen Geschwindigkeiten positioniert und bei Erreichen der Koinzidenz durch das Signal A=B der Antrieb abgeschaltet, so entsteht durch das Trägheitsmoment ein Überlauf, der u.U. die Gegenrichtung wieder einschaltet oder der bei Unterdrückung der Wiedereinschaltung als Positionierfehler stören kann. Bei Positionieraufgaben mit engem Toleranzbereich

(numerisch gesteuerte Werkzeugmaschinen, Roboter) ist es deshalb bei Annäherung der beiden Größen günstig, die Verfahrgeschwindigkeit proportional mit der Differenz zu verringern. Zur Lösung dieser Steuerungsaufgabe benötigt man die Subtraktion, wie sie auch als Grundrechenart in jedem Rechner und jeder Datenverarbeitungsanlage vorhanden ist.

Da jede kompliziertere Rechenoperation (Multiplikation, Division, Wurzel usw.) auf Folgen elementarer Operationen (Addition, Subtraktion, Verschiebung) zurückgeführt werden kann, hat der Adder als rein kombinatorische Schaltung eine herausragende Bedeutung erlangt. Dies um so mehr, da auch die Subtraktion durch eine Addition der Komplemente realisierbar ist. Deshalb wollen wir die Addition näher betrachten.

### 5.1.7.1. Das Rechnen mit binär kodierten Zahlen

Eingangs sei ausdrücklich gesagt: Beim Rechnen mit binär kodierten Zahlen gibt es gegenüber unserer bekannten, dezimalen Algebra keine neuen Rechenregeln, sondern nur Vereinfachungen, die sich ergeben, weil nur noch die Ziffern 0 und 1 existieren und damit jede Operation trivial wird.

#### 5.1.7.1.1. Dualzahlen

Beginnen wollen wir mit der Addition von Dualzahlen (Abb. 32, Spalte 2). Als Beispiel wollen wir 10 (1010) + 14 (1110) rechnen.

```
           1010
         + 1110
         ------
Übertrag  111
          11000
          =====
```

Das Ergebnis $1.2^4 + 1.2^3 + 0.2^2 + 0.2^1 + 0.2^0$
$1.16 + 1.8 = 24$

entspricht unserer Erwartung. Die einzige Schwierigkeit bei der Addition könnte im 1. Moment aufgetreten sein, wenn man 0 + 0 = 0 von rechts beginnend (niedrigste Wertigkeit)

gerechnet hat und danach für 1 + 1 = 2 schreiben möchte.
2 heißt aber dual 10, so daß die Summe 0 wird und ein Übertrag 1 entsteht. Es ergibt sich als vollständige Wertetabelle der Addition in Abb. 52.

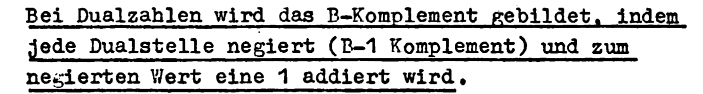
| Übertrag C | A | B | Übertrag $C_{+1}$ | Summe A+B $\Sigma$ |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 |

Abb. 52 Wertetabelle der Addition

Mit einem ähnlichen Vorgehen könnte man auch eine Wertetabelle für die Subtraktion aufstellen und später mit einer separaten Schaltung realisieren. Um aber den Schaltungsaufwand zu verringern, wird die Subtraktion fast immer durch die Addition des B-Komplementes ausgeführt. B bedeutet dabei die Basis des Zahlensystems. Im betrachteten Fall ist B = 2, weshalb in Dualrechnern teilweise vom 2er Komplement gesprochen wird. Komplement heißt die Ergänzung einer Teilmenge zur Gesamtmenge. Wenn beispielsweise ein Zahlenbereich aus $2^5 = 32$ Elementen betrachtet werden soll, dann ist das B-Komplement von 14 die Ergänzung zu 32.

$$32 - 14 = 18$$

18 ist beim angenommenen Zahlenbereich das B-Komplement zu 14. Für die automatische Bildung des Komplementes ist es nun wichtig, daß dies ohne Subtraktion möglich ist.



Bei Dualzahlen wird das B-Komplement gebildet, indem jede Dualstelle negiert (B-1 Komplement) und zum negierten Wert eine 1 addiert wird.

Bei der Subtraktion tritt erstmalig der Fall auf, daß das Ergebnis je nach Größe von Subtrahend und Subtraktor positiv

oder negativ werden kann. Es muß deshalb auch das Vorzeichen durch eine Bitstelle kodiert werden.

Merke: $+ \hat{=} 0$
$- \hat{=} 1$

Die Vorzeichenstelle wird durch einen Punkt vom Zahlenwert abgetrennt.

Praktisch sieht das so aus:

```
+ 14  =  0.01110
         ========
         1.10001 (Negation von + 14)
       +       1
         --------
- 14  =  1.10010
         ========
```

Am Beispiel 22 - 14 wollen wir die Richtigkeit überprüfen.

```
    22         0.10110
+ ( - 14)    + 1.10001
             ---------
              11  1111 (Übertrag)
  ------      --------
   + 8        0.01000
  ======      ========
```

Das Ergebnis 0.01000 entspricht der + 8. In der Rechnung haben wir die Addition der 1 zum negierten Ausdruck der + 14 gleich mit erledigt. Der Übertrag über die Vorzeichenstelle hinaus bleibt stets unberücksichtigt.

Abschließend wollen wir noch das Gegenbeispiel 14 - 22 probieren.

```
    14         0.01110
+ ( - 22)    + 1.01001
             ---------
 --------        11111
   - 8         1.11000
 ========      ========
```

Die 1 als Vorzeichen zeigt uns, daß die Differenz negativ ist. Damit steht aber nach dem Vorzeichen auch nicht der Betrag, sondern das Komplement des Betrages. Zur Überprüfung können wir den Betrag wiederum durch Komplementbildung erzeugen.

```
      1.11000
    =========
      0.00111
  +         1
    ---------
      0.01000
    =========
```

Wir sehen, daß die Differenz richtig ist.

### 5.1.7.1.2. BCD-Zahlen

Auch mit den BCD-Zahlen (Abb. 32, Spalte 3) können wir Addition und Subtraktion ausführen. Hier entsteht aber eine Schwierigkeit, die uns schon beim Addieren von Zeitangaben begegnet ist.

```
              3 h    45'
            + 2 h    25'
              ----------
                     1
            ------------
              5 h    70'
            ------------
Korrektur     6 h    10'
            ============
```

Da wir gewohnt sind, dezimal zu rechnen, der Minutenbereich aber bei 60 endet, muß korrigiert werden.

Im 8-4-2-1 Kode benutzen wir 4 Bit, um die dezimalen Ziffern 0......9 zu verschlüsseln. 4 Bit lassen aber 15 Kombinationen zu. Während wir im Fall unserer Zeitrechnung bei Minutenangaben >59 durch Addition von 40 die unzulässigen 40 Werte (60....99) übersprungen haben, müssen wir dies im 8-4-2-1 Kode durch Addition von 6 vornehmen. Alles übrige geschieht bei der Addition wie im Dualsystem.

<u>Bei Addition im 8-4-2-1 Kode muß in jeder Tetrade mit + 6 (0110) korrigiert werden, wenn die primäre Summe > 9 ist.</u>

```
Beispiel:    25          0010     0101
           + 18        + 0001     1000
                       ---------------
           ----        ---------------
             43          0011     1101
           ====      +   0000     0110   Korrekturaddition
                          111     1
                         0100     0011
                       ===============
```

Ein weiterer Nachteil des 8-4-2-1 Kodes ist bei der Subtrak - tion durch die Addition der B-Komplemente festzustellen. Da B in diesem Fall 10 ist, muß jede dezimale Ziffer zunächst zu B-1 ( ≙ 9 ) ergänzt werden und danach wie in Dualen eine 1 in der Stelle $10^0$ addiert werden. Die Ergänzung zu 9 ist <u>nicht</u> durch Negation aller Binärstellen möglich. Es wird eine besondere Logikschaltung benötigt. Mit einem Beispiel zur BCD-Subtraktion wollen wir die theoretische Betrachtung abschließen. Als weiterführende Literatur ist /4/ zu empfehlen.

```
    33            0011     0011
+ (- 15)        + 1000     0100   Ergänzung zu 9 je Tetrade
                           1111
--------        ---------------
   + 18           1011     1000
========        + 0110     0000   Korrektur bei > 9 mit 6
                 111
                ---------------
                  0001     1000
                ===============
```

## 5.1.7.2. Schaltungstechnik

Aus der Wertetabelle in Abb. 52 kann man sofort die logischen Gleichungen für Summe ( Σ ) und Übertrag zur nächsthöheren Binärstelle ($C_{+1}$) ablesen.

$$\Sigma = A \bar{B} \bar{C} \vee \bar{A} B \bar{C} \vee \bar{A} \bar{B} C \vee A B C \qquad (13)$$

$$C_{+1} = A B \bar{C} \vee A \bar{B} C \vee \bar{A} B C \vee A B C \qquad (14)$$

Bei Überprüfung mit dem Karnaugh-Plan stellen wir fest, daß sich nur die Gleichung 14 kürzer schreiben läßt. Wir erhalten

$$C_{+1} = A B \vee A C \vee B C \qquad (15)$$

In Abb. 53 sind die Gleichungen (13) und (15) eines Adders in NAND-Realisierung dargestellt. Diese Adderschaltung nennt man auch Volladder, da außer den zu addierenden Variablen ein möglicher Übertrag aus der vorhergehenden Stelle berücksichtigt wird. Bei der Addition in der niederwertigsten Stelle ist diese Anordnung nicht immer erforderlich. Eine Additionsanordnung

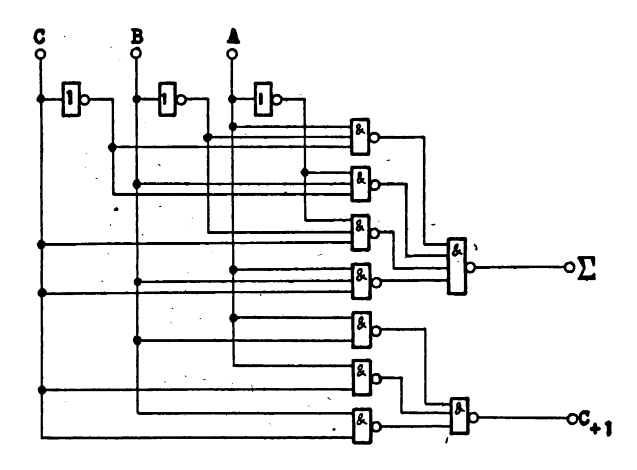



Abb. 53 Adderschaltung in 2-stufiger NAND-Technik

ohne Übertragsberücksichtigung heißt Halbadder. Sie verwirklicht die Gleichungen (16) und (17), die aus der Werttabelle der Abb. 52 entstehen, wenn C entfällt.

$$\Sigma' = \overline{A}\,B \;\mathrm{v}\; A\,\overline{B} \qquad (16)$$

$$C_{+1} = A\,B \qquad (17)$$

Zunächst erscheint es vielleicht günstig, für den Ausdruck $\Sigma'$ die UND - NOR - Verknüpfung des Schaltkreises D 150 einzusetzen. In diesem Fall müßten aber die Eingänge schon positiv und negiert zur Verfügung stehen. Für $C_{+1}$ würde aber trotzdem noch eine UND-Verknüpfung benötigt. Durch Umformung der Summengleichung kann man bei alleinigem Einsatz von NAND den Übertragsterm gleich mit erzeugen und dadurch eine bessere Elementeausnutzung erzielen. Ausgehend von Gleichung (16) erhalten wir durch Anwendung der Kürzungsregeln (LB 1 Abschnitt 3.1.) Gleichung (18)

$$\Sigma' = A\,(\,\overline{A} \;\mathrm{v}\; \overline{B}\,) \;\mathrm{v}\; B\,(\,\overline{B} \;\mathrm{v}\; \overline{A}\,) \qquad (18)$$

$$\Sigma' = A\;\overline{A\,B} \;\mathrm{v}\; B\;\overline{A\,B} \qquad (19)$$

$$\Sigma' = \overline{\overline{A\;\overline{A\,B}}\;\;\overline{B\;\overline{A\,B}}} \qquad (20)$$

Die Gleichung (20) benötigt nur noch D 100 Bausteine zur Realisierung und stellt die Teilsumme $\Sigma'$ und den negierten Teilübertrag $\overline{C_{+1}'}$ bereit.

Wenn nicht extreme Geschwindigkeitsforderungen eine 2-stufige Volladderschaltung verlangen, läßt sich mit einer Kettenschaltung von 2 Halbaddern aufwandsminimal die Summe $\Sigma$ bilden. Die Teilüberträge $C_{+1}'$ müssen dabei ODER verknüpft werden. Die Abb. 54 zeigt diese Realisierung. Beim Vergleich mit der Abb. 53 erkennt man, daß dort 1 x D 120, 2 x D 110 und 1 x D 100 den 2 1/4 x D 100 der 2. Lösung gegenüberstehen.

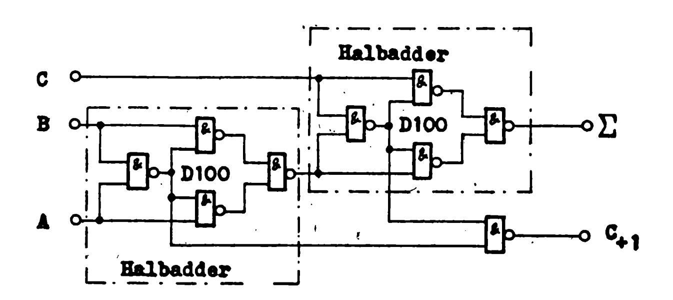



Abb. 54 Adderschaltung aus 2 Halbaddern

Da jedes NAND eine durchschnittliche Verzögerungszeit von 12 ns bringt, ist in der Variante nach Abb. 54 zu beachten, daß in parallelen Adderanordnungen für größere Wortbreiten der Übertrag unbedingt an den jeweils 2. Halbadder angelegt wird. Trotzdem muß schon bei 4-Bit-paralleler Verarbeitung 12 Verzögerungszeiten auf das Ergebnis gewartet werden, wenn nicht besondere Schaltungsmaßnahmen getroffen werden. In mittelintegrierten Adderschaltkreisen z.B. 4 Bit Adder SN 7483 bzw. K 155 IM 3 wird der Übertrag durch eine spezielle Schaltung generiert, so daß für 4 Bit nur 16 ns Verzögerungszeit auftreten.

Zum Abschluß sei als Applikation noch eine Adderanordnung betrachtet, die durch ein Steuersignal (S) zwischen dualer Addition (S = L) und dualer Subtraktion (S = H → A - B) umgeschaltet werden kann. Dabei wird auch für die Subtraktion garantiert, daß unabhängig von Subtrahend und Subtraktor ausgangsseitig

der Betrag der Differenz bereitsteht (Abb. 55).

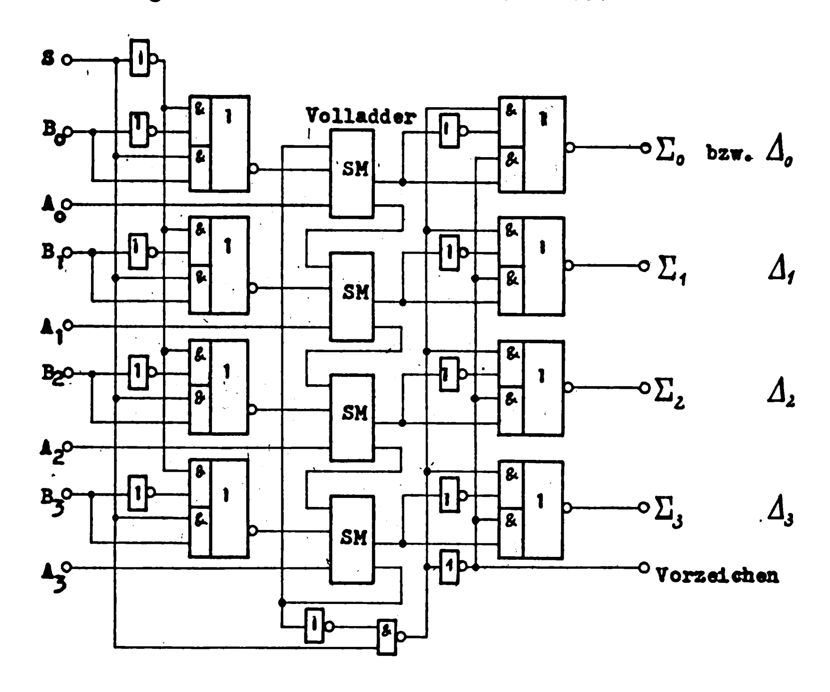



Abb. 55 4 Bit Additions- und Subtraktionsschaltung mit Betragsbildung

Das Beispiel ist für eine Wortbreite von 4 Bit dargestellt. Es kann beliebig erweitert werden. Als Adder können Anordnungen nach Abb. 53 und 54 oder integrierte Schaltkreise eingesetzt werden. Die D 150 vor dem Adder bilden das B-1-Komplement bei Subtraktion, die D 150 nach dem Adder übernehmen die Rückkomplementierung bei negativer Differenz. Gleichzeitig wird der Übertragseingang an der Stelle $2^0$ vom Übertragsausgang so gesteuert, daß nur bei positiver Differenz das B-1-Komplement zum B-Komplement ergänzt wird. Als Steuersignal S kann in der vorliegenden Schaltung das Vorzeichen der Eingangsgröße B benutzt werden. Der Funktionsumfang der Schaltung (Abb. 55) wird im TTL-Sortiment in integrierter Form (SN 74181 bzw. 74381) angeboten und ermöglicht damit aufwandsarme Realisierungen.

## Fragen zum Abschnitt 5.1.

### Teil 1 - Versuche mit dem Lehrbaukasten

1. Stecken Sie die Schaltung eines $\binom{4}{1}$ Dekoders nach Abb. 31! Benutzen Sie zur Erzeugung der Eingangsbelegungen 2 Tastenschalter (9), und kontrollieren Sie mit 4 Anzeigelampen (4) die Schaltbelegungstabelle!

2. Erweitern Sie die Schaltung aus Aufgabe 1 durch einen gemeinsamen Strobe-Eingang! Überprüfen Sie die Funktion!

3. Für eine spezielle Einrichtung sind für die Signale A....E folgende Kodierungen zu erzeugen:

| | Y4 | Y3 | Y2 | Y1 |
|---|---|---|---|---|
| A | 0 | 0 | 1 | 1 |
| B | 0 | 1 | 0 | 1 |
| C | 0 | 1 | 1 | 0 |
| D | 1 | 0 | 1 | 0 |
| E | 1 | 1 | 0 | 0 |

   Entwerfen Sie einen Kodierer mit NAND-Bausteinen, und überprüfen Sie die Schaltung!

4. Entwerfen Sie eine Schaltung mit NAND-Elementen für 3 Eingangsvariable, die genau dann Eins wird, wenn eine ungerade Anzahl von Eingängen Eins ist.

5. Bauen Sie einen Volladder aus 2 Halbaddern auf, und überprüfen Sie dessen Funktion!

6. Es sind 2 Worte A ($A_1A_0$) und B ($B_1B_0$) zu vergleichen. (Benutzen Sie die 2 linken und die 2 rechten Tastenschalter (9)). Die Schaltung soll nur ein Signal erzeugen, wenn $A > B$ gilt. Entwerfen Sie die Schaltung mit NAND-Elementen und überprüfen Sie die Funktion.

### Teil 2 - Formale Aufgaben

7. Errechnen Sie nach Formel (6) wieviel Variable mindestens benötigt werden, um $3 \cdot 10^7$ Zustände zu kodieren.

8. Errechnen Sie, wieviel Bit theoretisch zur binären Kodierung einer Dezimalstelle ausreichen! Wieviel Prozent sind das gegenüber den praktisch benötigten 4 Bit der BCD-Darstellung?

9. Entscheiden Sie nur nach ökonomischen Gesichtspunkten, ob unter der Annahme, daß 4 Eingangsvariable vorhanden sind und alle 16 Kombinationen auftreten können, für das Erkennen der Zustände 0, 5, 9, 10, 15 (low aktiv) eine Realisierung mit den benötigten niedrig integrierten Schaltkreisen aus dem Spektrum D 100....D 130 billiger wird als ein mittelintegrierter $\binom{16}{1}$ Dekoder, wenn dieser den 3,5fachen Preis eines NAND-Schaltkreises ausmacht!

10. Für welche Variante entscheiden Sie sich, wenn für das Problem aus Aufgabe 3 eine Lösung gesucht wird und zusätzlich bekannt ist, daß die Eingangsvariablen positiv und negiert verfügbar sind und der Zustand 13 eingangsseitig nicht auftreten kann?

11. Wieviel Signale kann man mit integrierten TTL-Schaltungen der Grundreihe D 100...D 130 <u>einstufig</u> dual kodieren, wenn die Eingänge positiv und negiert vorhanden sind?

12. Es liegt ein Zahlenbereich 0....63 dual kodiert vor. Mit der Grundreihe D 100.....D 130 ist eine Schaltung mit möglichst geringer Kettenlänge zu entwerfen, die ein Signal bei den Zuständen erzeugt, wo die Quersumme der Einsen eine gerade Zahl ist (Paritätsprüfung). Wieviel IS werden insgesamt benötigt, wenn die für die Schaltung positiv und negiert verfügbaren Eingangsgrößen außerhalb schon mit 8 TTL-Lasten belegt sind?

13. Als Beispiel eines Umkodierers ist das Funktionsbündel zur Bildung von B-1 Komplementen bei binär kodierten Dezimalstellen (BCD) zu ermitteln!

Randbedingung: Eingangsseitig treten keine Pseudotetraden auf.

14. Geben Sie die minimierte Gleichung an, die sicherstellt, daß bei BCD-Addition und einem Ergebnis > 9 für die Korrekturaddition in dieser Tetrade das Aufrufsignal der +6 (0110) bereitsteht.

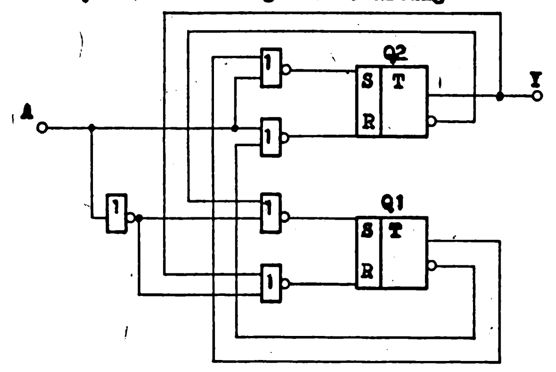
15. Der Übertragseingang eines Volladders ist logisch gleichwertig den übrigen Variableneingängen.
Im Abschnitt 5.1.7. ist die maximale Signallaufzeit für eine 4-Bit-breite Adderanordnung aus Halbaddern mit 12 Verzögerungszeiten angegeben. Wie groß ist die maximale Laufzeit, wenn die A-Eingänge als Übertragseingänge benutzt werden und der zu addierende Zahlenwert an die frei gewordenen Übertragseingänge (C) gelegt wird?
(1 Verzögerungszeit = Verzögerungszeit eines logischen Gatters.)

16. Stellen Sie eine Wertetabelle zur direkten Subtraktion auf, und vergleichen Sie die logische Gleichung des Differenzsignales mit dem Summensignal eines Adders!
Vergleichen Sie auch die Übertragssignale!

## Anhang

### 2. Analysebeispiel

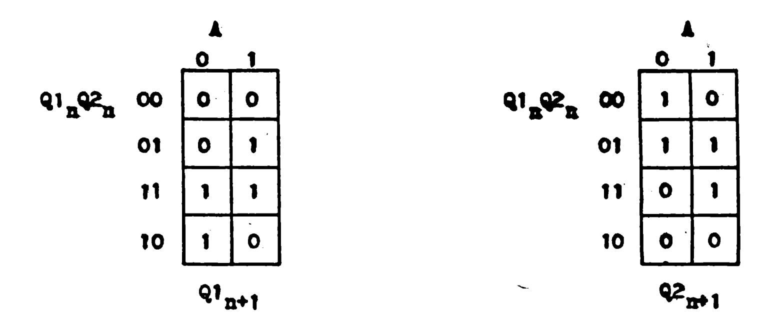
Analysieren Sie folgende Schaltung




1. Schritt: Ist trivial, da konzentrierte Speicherelemente
2. Schritt: Konstituententafeln

$S2 = \overline{A \vee Q1} = \bar{A}\,\overline{Q1}$ $\qquad S1 = \overline{\overline{Q2} \vee \bar{A}} = A\,Q2$ $\qquad Y = Q2$

$R2 = \overline{A \vee \overline{Q1}} = \bar{A}\,Q1$ $\qquad R1 = \overline{Q2 \vee \bar{A}} = A\,\overline{Q2}$



$Q1_{n+1}$:

| $Q1_n Q2_n$ | A = 0 | A = 1 |
|---|---|---|
| 00 | 0 | 0 |
| 01 | 0 | 1 |
| 11 | 1 | 1 |
| 10 | 1 | 0 |

$Q2_{n+1}$:

| $Q1_n Q2_n$ | A = 0 | A = 1 |
|---|---|---|
| 00 | 1 | 0 |
| 01 | 1 | 1 |
| 11 | 0 | 1 |
| 10 | 0 | 0 |

Beachte: Die durch S und R nicht beschriebenen Felder betreffen den Speicherzustand und sind selbständig einzutragen.

3. Schritt: Wirktafel

| Q1 Q2 | A = 0 | A = 1 |
|---|---|---|
| 00 | 2 | (1) |
| 01 | (2) | 3 |
| 11 | 4 | (3) |
| 10 | (4) | 1 |

4. Schritt: Wirkliste

| A = 0 | A = 1 | Y |
|---|---|---|
| 2 | (1) | 0 |
| (2) | 3 | 1 |
| 4 | (3) | 1 |
| (4) | 1 | 0 |

5. Schritt: Impulsdiagramm

A

Q1

Q2

Y

Frequenzhalbierglied

Lösungen zu den Aufgaben 4.4.

1. -

2. $Q_{n+1} = \bar{B}\,(A \vee Q_n) = A\,\bar{B} \vee \bar{B}Q_n$

$= \overline{\overline{A\,\bar{B}}\;\;\overline{\bar{B}Q_n}}$

3. Es wird ein JK-Flip-Flop verwendet, bei dem J = K = H geschaltet ist.
Das Tastersignal wird an den Takteingängen gelegt.
(betätigter Taster ≙ H)
(unbetätigter Taster ≙ L)
Der Ausgang Q entspricht dem Lampensignal.

4. -

5. -

6. -

7. Mit $\bar{R}$ = H wird der Speicher gesetzt
Mit $\bar{S}$ = H wird der Speicher rückgesetzt

8. Frequenzdrittelung

9. a) Der Zustand beider Speicher ist identisch
b) Der Zustand des 1. Speicherelementes wird bei der H/L-Flanke des Taktes in den 2. Speicher übertragen (Master-Slave-Verhalten).

10. Es wird ein Speicherelement eingesetzt.
Bei einer Zuordnung der Füllstandsmeßeinrichtungen
0 ≙ Pegel unterschritten
1 ≙ Pegel überschritten
und der Zuordnung A ≙ oberer- B ≙ unterer Füllstand ergibt sich:

$S = \bar{B} \qquad R = A\,B$

11. Die reduzierte Wirkliste enthält nur noch 2 Zeilen, d.h., ein Speicherelement ist einzusetzen.

$S = A\,B \qquad R = \bar{A}\,\bar{B}$

12. Die vier Zeilen der Wirkliste lassen sich nicht reduzieren. Die FF zählen aber in der Folge 0-1-3-2-0 usw. Es ist eine Umkodierung notwendig.

$SQ_1 = \bar{A}\, Q_2 \qquad SQ_2 = A\, \overline{Q_1} \qquad Y1 = \overline{Q_1} Q_2 \vee Q_1 \overline{Q_2}$

$RQ_1 = \bar{A}\, \overline{Q_2} \qquad RQ_2 = A\, Q_1 \qquad Y2 = Q_2$

Index 1 ≙ niederwertige Stelle des Zählers bzw. Ausgangs,
Index 2 ≙ höherwertige Stelle des Zählers bzw. Ausgangs.

Antworten zu den Fragen des Abschnittes 5.1.

1. -

2. -

3. $Y1 = \overline{\bar{A}\ \bar{B}}$
   $Y2 = \overline{\bar{A}\ \bar{C}\ \bar{D}}$
   $Y3 = \overline{\bar{B}\ \bar{C}\ \bar{E}}$
   $Y4 = \overline{\bar{D}\ \bar{E}}$

4. $Y = A\bar{B}\bar{C} \vee \bar{A}B\bar{C} \vee \bar{A}\bar{B}C \vee ABC$
   Y entspricht damit genau der Summe Σ des Volladders. Zur Verkürzung der Schreibweise solcher Ausdrücke findet man in der Literatur auch $Y = \binom{3}{1} \vee \binom{3}{3}$

5. -

6. Durch Minimierung mit dem Karnaugh-Plan findet man für 2-Bit-breite Worte A bzw. B.
   $A > B = A_1 \bar{B}_1 \vee A_1 A_0 \bar{B}_0 \vee A_0 \bar{B}_1 \bar{B}_0$

7. $n \geq 3{,}32 . \lg 3 + 7 \lg 10$
   $n \geq 24{,}8$
   $n = 25$

8. $n \geq 3{,}32 \cdot \lg 10$

$n = 3{,}32$

3,32 Bit ≙ 83 % von 4 Bit

9. Für die Dekodierung gelten ohne Randbedingungen folgende Gleichungen:

$$0 = \overline{\bar{D}\,\bar{C}\,\bar{B}\,\bar{A}}$$
$$5 = \overline{\bar{D}\,C\,\bar{B}\,A}$$
$$9 = \overline{D\,\bar{C}\,\bar{B}\,A}$$
$$10 = \overline{D\,\bar{C}\,B\,\bar{A}}$$
$$15 = \overline{D\,C\,B\,A}$$

Als Aufwand ergibt sich:

1 x D 100 (Negierung der Eingangsgrößen)
3 x D 120 (5 x NAND mit 4 Eingängen)
4 IS

Entscheidung zugunsten $\binom{16}{1}$ Dekoders!

10.

$$0 = \overline{\bar{D}\,\bar{C}\,\bar{B}\,\bar{A}}$$
$$5 = \overline{C\,\bar{B}\,A}$$
$$9 = \overline{D\,\bar{B}\,A}$$
$$10 = \overline{D\,\bar{C}\,B\,\bar{A}}$$
$$15 = \overline{D\,C\,A}$$

Aufwand: 1 x D 110
1 x D 120
2 IS

Entscheidung zugunsten der niedrig integrierten Lösung!

11. Maximal 16 Signale sind einstufig kodierbar, da beim Dualkode 0 und 1 gleich verteilt ist und mit dem D 130 8 negierte Signale als ODER verarbeitet werden.

12. Bei 64 Zuständen sind 6 Bit vorhanden. Die Funktion Y für alle Zustände mit gerader Quersumme lautet verkürzt geschrieben

$$Y = \binom{6}{0} \vee \binom{6}{2} \vee \binom{6}{4} \vee \binom{6}{6}$$

Y wird damit durch ODER aus 32 UND Gliedern mit je 6 Eingängen gebildet. Da jedes Bit sowohl positiv als auch negiert an 16 UND-Eingänge angeschlossen werden muß, ist die Entkopplung über je 2 TTL-Elemente erforderlich. Für die ODER-Funktion stehen nur NAND mit 8 Eingängen bereit, so daß in 2 Stufen mit zwischengeschalteter Negierung zusammengefaßt werden muß.

| | | | |
|---|---|---|---|
| 6 | x | D 100 | (Entkopplung) |
| 32 | x | D 130 | UND_Glieder |
| 4 | x | D 130 | ODER-Funktion |
| 1 | x | D 100 | |
| 1 | x | D 120 | |
| 44 | | IS | |

13.

| D $2^3$ | C $2^2$ | B $2^1$ | A $2^0$ | Y3 | Y2 | Y1 | Y0 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 |
| 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 |
| 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |

Unter Berücksichtigung der Ø Felder erhält man:

$Y0 = \overline{A}$ $\qquad$ $Y1 = B$

$Y2 = C\,\overline{B} \vee \overline{C}\,B$

$Y3 \quad \overline{D}\,\overline{C}\,\overline{B}$

14. Der Tetradenübertrag sei $C_{3+1}$ genannt, die Summenausgänge $\Sigma 3 \ldots \Sigma 0$. Für die Gleichung ergibt sich dann

$$Y = C_{3+1} \vee \Sigma 3\ \Sigma 2 \vee \Sigma 3\ \Sigma 1$$

15. Je Volladder (bestehend aus 2 Halbaddern) durchläuft das Summensignal 6 Gatter. Zwischen Übertragseingang und Übertragsausgang eines Volladders liegen 2 Gatter.
Zwischen Übertragseingang und Summenausgang befinden sich 3 Gatter. Für den schnellen Fall ergibt sich:

$$\text{Summenlaufzeit} = 3 + 2 + 2 + 2 + 3 = \underline{\underline{12}}$$

Für den langsamen Fall ergibt sich:

$$\text{Summenlaufzeit} = 3 + 2 + 3 + 2 + 3 + 2 + 3 + 3 = \underline{\underline{21}}$$

16. Für die Subtraktion $A - B = \Delta$ ergibt sich

| A | B | M | $M_{+1}$ | $\Delta$ |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 |

M = Mangel

Der Vergleich zeigt $\Delta = \Sigma$, aber für $M_{+1}$ ergibt sich mit $M_{+1} = \bar{A}\,B \vee \bar{A}\,M \vee B\,M$ ein Ausdruck der mit einer Schaltung zur Generierung des Übertrages $C_{+1}$ nur erzeugt werden kann, wenn statt A die Negation von A angelegt wird.

Ergänzung der Liste der Bausteine des Lehrbaukastens

| Bezeichnung | Symbol | Funktion | Anzahl |
|---|---|---|---|
| 2 x NAND mit 2 Eingängen Kollektor offen | A, B → & → Y; A, B → & → Y | $Y = \overline{A \wedge B}$ mit externem Widerstand | 2 |
| BCD zu 7 - Segment-Dekoder | A, B, C, D, RB, BI → 1, 2, 4, 8, RB, BI; x/y; A–G → a–g; RB | | 2 |
| D-Flip-Flop | S, D, T, R → S, D, C, R; T; Q, $\overline{Q}$ | $Q_{n+1} = D(tn)$ | 2 |
| J-K-Master-Slave Flip-Flop | $\overline{S}$, J1, J2, J3, T, K1, K2, K3, $\overline{R}$ → S, &, J, C, &, K, R; Q, $\overline{Q}$ | $Q = J\overline{Qn} \vee \overline{K}Qn$ | 2 |
| Verteilerfeld | | | 1 |

| Bezeichnung | Symbol | Funktion | Anzahl |
|---|---|---|---|
| 2 x<br>Volladder | A, B, C; S1, S2, $Ü_E$; SM; S, $Ü_A$; Σ, C+1<br>A, B, C; S1, S2, $Ü_E$; SM; S, $Ü_A$; Σ, C+1 | $\Sigma = \binom{3}{1} \vee \binom{3}{3}$<br>$C+1 = \binom{3}{2} \vee \binom{3}{3}$ | 2 |
| Dekoder mit Strobeeingang<br>$\binom{4}{1}$ | A, B, Str; 1, 2, Str; DC; 0, 1, 2, 3; Y0, Y1, Y2, Y3 | $Y0 = \bar{A}\ \bar{B}\ \overline{Str}$<br>$Y1 = A\ \bar{B}\ \overline{Str}$<br>$Y2 = \bar{A}\ B\ \overline{Str}$<br>$Y3 = A\ B\ \overline{Str}$ | 1 |
| Multiplexer<br>4 zu 1 | A, B; 1, 2; DC; C, D, E, F; MPX; Y | $Y = C0 \vee D1 \vee E2 \vee F3$ | 1 |

## Literaturverzeichnis


/1/ Leonhardt, E.: Grundlagen der Digitaltechnik, Verlag Technik, Berlin 1977.

/2/ anonym Bipolare digitale Schaltkreise, Halbleiterwerk Frankfurt/Oder Katalog 1975/1976.

/3/ anonym The TTL - Data - Book Supplement to CC - 401 TI 1973.

/4/ Murphy, J.S.: Elektronische Ziffernrechner, Verlag Technik, Berlin 1965.

/5/ Bochmann, D.: Einführung in die strukturelle Automatentheorie. Verlag Technik, Berlin 1975.

/6/ Böhringer, M.: Theorie und Technik von Schaltnetzwerken, Verlag Technik, Berlin 1969.